



HAL
open science

Chance Constrained Programming: with applications in Energy Management

Wim van Ackooij

► **To cite this version:**

Wim van Ackooij. Chance Constrained Programming: with applications in Energy Management. Other. Ecole Centrale Paris, 2013. English. NNT : 2013ECAP0071 . tel-00978519

HAL Id: tel-00978519

<https://theses.hal.science/tel-00978519>

Submitted on 14 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ÉCOLE CENTRALE DES ARTS
ET MANUFACTURES
«ÉCOLE CENTRALE PARIS»

THÈSE
présentée par

Wim Stefanus van Ackooij

pour l'obtention du

GRADE DE DOCTEUR

Spécialité: Applied Mathematics

Laboratoire d'accueil: LGI

SUJET:

Chance Constrained Programming
with applications in Energy Management

soutenu le : 12 December 2013

devant un jury composé de:

G. Oppenheim	Président
W. Römisich	Rapporteur
C. Lemaréchal	Rapporteur
V. Mousseau	Examineur
M. Minoux	Directeur de Thèse
R. Henrion	Co-Directeur de Thèse

This thesis is dedicated to
my wife Séverine and children Elsa, Charlotte and Tom

Jakobus 1: 5. "Indien echter iemand van u in wijsheid te kort schiet, dan bidde hij God daarom, die aan allen geeft, eenvoudigweg en zonder verwijt; en zij zal hem gegeven worden"

Acknowledgements

In a way the basis for this thesis was laid in 2005 when together with some colleagues we were quantifying the impact of uncertainty on weekly unit-commitment. A fruitful collaboration with Pierre Thomas and Riadh Zorgati led to the birth of a software package called LEA ("Laboratoire des Expérimentations des Aléas") in the same year. From a technical perspective this software package allowed the user to execute a simple multi-scenario approach around a deterministic optimization kernel. The innovating part was set up with Bernard Beauzamy. Instead of drawing scenarios as iid samples from some law (or black box software), the idea was to build artificial englobing scenarios allowing for bounding of output with only little computational effort. These experiments led, only much later, to an industrial software package. In any case this is how together with Riadh Zorgati, I got interested in dealing with uncertainty in an appropriate way. We had many discussions on Scenario generation, Robust Optimization etc These discussions were also greatly nourished by discussions with Michel Minoux. Around the same time, we had the chance to meet René Henrion and attend a seminar he gave. Chance constrained programming looked like a very powerful and intuitive way to deal with uncertainty. With good hopes and enthusiasm we set out to look where uncertainty in Energy Management plays a key role and should be dealt with appropriately. Once these structures identified we started experimenting with safe tractable approximations of chance constraints, scenario type of approaches and also Robust Optimization. Somewhere in 2009 we had to chance to work closely together with René Henrion and look to what extent Chance Constrained programming could be applied in Energy Management. This led me to be greatly intrigued by this field, and ultimately, gave me a desire to deeply understand this domain. Thanks to EDF I have had the opportunity to do this Thesis in part time. I hope to have contributed to the domain and continue to do so in the future. I do firmly believe that Chance Constrained Programming will be important in Energy Management in the near future.

Thanking people after 10 years of work at EDF is almost a dangerous undertaking since we are bound to forget someone with whom we have had interesting discussions. Nonetheless I will make this attempt at my own risks.

First of all I would like to thank Riadh Zorgati, who is a good friend and colleague. We share many views and in particular that (fundamental) research should have its place in an(y) industrial company. I would also like to thank Grace Doukopoulos, Olivier Feron, Nadia Oudjane, Cyrille Strugarek and Xavier Warin for the many interesting discussions (and years of joint work). Thanks to Thomas Triboulet, Laurent Mulot and their pragmatic viewpoint of what at EDF we commonly call "le métier" it is possible to obtain a partial view of the underlying structures of the optimization problems at hand. This allows for important progress in identifying the key features of Energy Management.

Clearly there is not only Chance Constrained Programming in the world, but also Aikido. Thanks go out to my friend Damien Jacomy, senseis Christian Mouza, Nicolas Paillat and François Pichereau. Aikido provides many physical (and intellectual) challenges in the (eternal) quest for perfected techniques. In a way it is pretty much like Mathematics, where elegance is also looked for.

I also naturally would like to thank René Henrion, Michel Minoux, Claudia Sagastizábal, Jérôme Malick, Antonio Frangioni, Welington de Oliveira for the joint work and/or important discussions. I am also grateful that Claude Lemaréchal and Werner Römisch have accepted to referee this Thesis.

Finally, I would like to thank my wife Séverine (and children Elsa, Charlotte and Tom) for all their love and support. Since this is a work on "robust" optimization, I should maybe thank all my future children as well. One never knows!

Abstract

In optimization problems involving uncertainty, probabilistic constraints are an important tool for defining safety of decisions. In Energy management, many optimization problems have some underlying uncertainty. In particular this is the case of unit commitment problems. In this Thesis, we will investigate probabilistic constraints from a theoretical, algorithmic and applicative point of view. We provide new insights on differentiability of probabilistic constraints and on convexity results of feasible sets. New variants of bundle methods, both of proximal and level type, specially tailored for convex optimization under probabilistic constraints, are given and convergence shown. Both methods explicitly deal with evaluation errors in both the gradient and value of the probabilistic constraint. We also look at two applications from energy management: cascaded reservoir management with uncertainty on inflows and unit commitment with uncertainty on customer load. In both applications uncertainty is dealt with through the use of probabilistic constraints. The presented numerical results seem to indicate the feasibility of solving an optimization problem with a joint probabilistic constraint on a system having up to 200 constraints. This is roughly the order of magnitude needed in the applications.

The differentiability results involve probabilistic constraints on uncertain linear and non-linear inequality systems. In the latter case a convexity structure in the underlying uncertainty vector is required. The uncertainty vector is assumed to have a multivariate Gaussian or Student law. The provided gradient formulae allow for efficient numerical sampling schemes. For probabilistic constraints that can be rewritten through the use of Copulae, we provide new insights on convexity of the feasible set. These results require a generalized concavity structure of the Copulae, the marginal distribution functions of the underlying random vector and of the underlying inequality system. These generalized concavity properties may hold only on specific sets.

Les contraintes en probabilité constituent un modèle pertinent pour gérer les incertitudes dans les problèmes de décision. En management d'énergie de nombreux problèmes d'optimisation ont des incertitudes sous-jacentes. En particulier c'est le cas des problèmes de gestion de la production au court-terme. Dans cette Thèse, nous investiguons les contraintes probabilistes sous l'angle théorique, algorithmique et applicative. Nous donnons quelques nouveaux résultats de différentiabilité des contraintes en probabilité et de convexité des ensembles admissibles. Des nouvelles variantes des méthodes de faisceaux "proximales" et "de niveaux" sont spécialement mises au point pour traiter des problèmes d'optimisation convexe sous contrainte en probabilité. Ces algorithmes gèrent en particulier, les erreurs d'évaluation de la contrainte en probabilité, ainsi que son gradient. La convergence vers une solution du problème est montrée. Enfin, nous examinons deux applications : l'optimisation d'une vallée hydraulique sous incertitude sur les apports et l'optimisation d'un planning de production sous incertitude sur la demande. Dans les deux cas nous utilisons une contrainte en probabilité pour gérer les incertitudes. Les résultats numériques présentés semblent montrer la faisabilité de résoudre des problèmes d'optimisation avec une contrainte en probabilité jointe portant sur un système de environ 200 contraintes. Il s'agit de l'ordre de grandeur nécessaire pour les applications.

Les nouveaux résultats de différentiabilité concernent à la fois des contraintes en probabilité portant sur des systèmes linéaires et non-linéaires. Dans le deuxième cas, la convexité dans l'argument représentant le vecteur incertain est requise. Ce vecteur est supposé suivre une loi Gaussienne ou Student multi-variée. Les formules de gradient permettent l'application directe d'un schéma d'évaluation numérique efficient. Pour les contraintes en probabilité qui peuvent se réécrire à l'aide d'une Copule, nous donnons de nouveaux résultats de convexité pour l'ensemble admissibles. Ces résultats requièrent la concavité généralisée de la Copule, les distributions marginales sous-jacentes et du système d'incertitude. Il est suffisant que ces propriétés de concavité généralisée tiennent sur un ensemble spécifique.

List of Notation

1. \mathbb{R}^n : The set of n dimensional real numbers
2. \mathbb{P} : A probability measure
3. $\mathbb{E}(\cdot)$: The expectation operator
4. $\|\cdot\|$: A norm on \mathbb{R}^n , usually the Euclidian norm.
5. m_α : The mapping defining generalized Concavity (see Definition 2.5.3).
6. $M(p)$: The feasible set of a chance constrained decision vector (e.g., (2.2.1))
7. \cdot^\top : The transpose of a matrix or vector
8. $\text{conv}(M)$: The convex hull of the set M .
9. $\ker A$: The null-space of the linear operator A .
10. $\mathbb{R}^{m \times n}$: The set of $m \times n$ real matrices.

Contents

Abstract	i
List of Notation	iii
1 Introduction	1
1.1 Introduction	1
1.2 Main Contribution of this Work	3
2 Preliminaries	5
2.1 Measurability	6
2.2 Closedness of the Feasible Set	7
2.3 Continuity of Probabilistic Constraints	9
2.4 Connectedness of Feasible Set	10
2.5 Convexity of the Feasible Set	10
2.5.1 Special Cases with Convexity	14
2.5.2 Eventual Convexity of the Feasible set	15
2.6 (Local) Lipschitz Continuity	16
2.7 Differentiability of Probabilistic Constraints	16
2.7.1 General Statements	17
2.7.2 Special Cases	18
2.8 Algorithms	20
2.8.1 Feasible direction Methods	21
2.8.2 SUMT or "penalty function methods"	21
2.8.3 Interior Point Methods	21
2.8.4 Supporting Hyperplane Method	22
2.9 Related Concepts	22

2.9.1	Stability	22
2.9.2	p -Efficient points	23
2.9.3	Safe Tractable Approximations	23
2.9.4	Scenario Approximations / Sample Average Approximations	24
3	Theoretical Extensions	26
3.1	Semi-Infinite Individual Chance Constraints	26
3.2	Efficient Gradient Formulae : bilateral separable probabilistic constraints	31
3.3	Efficient Gradient Formulae : nonlinear probabilistic constraints with Gaussian and Gaussian-like distributions	37
3.3.1	A gradient formula for parameter-dependent Gaussian probabilities in the convex case	39
3.3.2	Selected Examples	50
3.3.2.1	Gaussian distributions	50
3.3.3	Gaussian-like distributions	51
3.3.4	Student (or T-) distribution	54
3.4	Eventual Convexity	61
3.4.1	Notation	62
3.4.2	Copulae and generalized concavity	63
3.4.2.1	Structure of the family of δ - γ -concave Copulae	64
3.4.2.2	Tools for deriving δ - γ -Concavity of Copulae	66
3.4.2.3	Estimates with Copulae	66
3.4.3	Eventual Convexity of the Feasible Set	68
3.4.4	A generalization of Results	71
3.4.4.1	Improved use of Generalized Concavity of the Constraint Mappings	71
3.4.4.2	Improved Estimates of p^*	72
3.4.4.3	More Copulae	74
3.4.5	A Partial Characterization of the Gaussian Copula	77
3.4.6	A Potential Application	80
4	Algorithms for (convex) Probabilistic Programming	82
4.1	A proximal Bundle Method	83
4.1.1	Designing Bundle methods for Constrained Optimization	85

4.1.1.1	Initial setting	85
4.1.1.2	Handling inexact oracle information	87
4.1.1.3	A Non-smooth Optimization Solver for Inexact Oracles	89
4.1.2	Asymptotic Analysis	91
4.1.3	Link with the Original Problem	97
4.1.4	Convergence Results	97
4.1.4.1	Relation with previous work	100
4.1.5	Energy Application: Hydro Reservoir Management	102
4.1.5.1	Devising an inexact upper oracle for the constraint	102
4.1.5.2	Convergence results for the application	104
4.1.6	Numerical experience	105
4.1.6.1	The various compared algorithms	106
4.1.6.2	Computational results	108
4.1.6.3	Feasible Start using a Slater Point and Infeasible Start	109
4.1.6.4	Different variants of Alg.PB.SEV	111
4.1.7	Solution Quality	112
4.2	A level Bundle Method	114
4.2.1	Some examples coming from stochastic optimization	116
4.2.2	Level Bundle Method for Exact Oracles	117
4.2.2.1	Convergence analysis	123
4.2.2.2	Complexity results	127
4.2.3	Level Bundle Method for Inexact Oracles	128
4.2.3.1	Inexact constrained level methods for upper oracles	130
4.2.3.2	Inexact constrained level methods for upper oracles with on-demand accuracy	131
4.2.3.3	Inexact constrained level methods for lower oracles	132
4.2.3.4	Constrained level bundle methods for lower oracles with on-demand accuracy	134
4.2.4	Numerical experiments	137
4.2.4.1	Benchmark Instance	137
4.2.4.2	Preconditioning	138
4.2.4.3	The various compared algorithms	139
4.2.4.4	An extensive discussion of numerical results	139

4.2.4.5	Comments on Parameter Settings	145
5	Applications of JCCP in large scale Unit-Commitment through decomposition	147
5.1	Robust Cascaded Reservoir Management	148
5.1.1	Problem Description	149
5.1.1.1	Problem Constraints	150
5.1.1.2	Objective function	156
5.1.1.3	Matrix formulation	156
5.1.2	Models for dealing with uncertainty	157
5.1.2.1	Expectation model	157
5.1.2.2	A Joint Chance Constraint Model (JCCP)	158
5.1.2.3	Max-P Problem	160
5.1.2.4	Individual Chance Constraint Model (ICCP)	160
5.1.2.5	A Robust Model	160
5.1.3	Numerical Example	162
5.2	Robust Unit-Commitment	165
5.2.1	Problem Structure	167
5.2.1.1	Hydro-Thermal Unit-Commitment Problems	167
5.2.1.2	A bird's view of the structure	169
5.2.2	Decomposition Methods	170
5.2.2.1	Lift and Dualize	171
5.2.2.2	Lift and Dualize - Augmented Lagrangian	172
5.2.2.3	Diagonal Quadratic Approximation	175
5.2.3	Algorithmic and Numerical Considerations	175
5.2.3.1	Slater Points	175
5.2.3.2	Algorithmic Considerations	177
5.2.4	Numerical Example: a (simplified) Robust Unit-Commitment Problem	178
5.2.5	Robust System-Dispatch Problem	178
5.2.5.1	The Thermal Generation Unit sub-problem	179
5.2.5.2	Offer Demand Equilibrium constraint	179
5.2.6	Numerical Example: Data and Results	180

5.2.6.1	Numerical Data	180
5.2.6.2	Numerical results	180
5.2.6.3	An extensive discussion of Numerical results	181
6	Concluding Remarks	189
	Bibliography	191

List of Figures

3.1	Illustration of solving problem (3.3.38) with the derived gradient formula. The right figure provides a zoom near the optimal solution.	60
4.1	Numerical Instance Data	103
4.2	Evolution of reservoir “Saut Mortier”, for the expected-value estrategy (left) and for Alg.PB.SEV strategy (right).	113
4.3	Comparison of Algorithm 4.2 when solving LP (4.2.17).	142
5.1	Trajectories of filling levels in reservoir ”Saut Mortier” and instance 2 for 100 simulated inflow scenarios. From top left to bottom right, solutions of problems (5.1.24), (5.1.25), (5.1.27), (5.1.28), (5.1.28)-Calib and (5.1.26) .	164
5.2	Turbined volumes (m^3) for instance 2. From top left to bottom right, solutions of problems (5.1.24), (5.1.25), (5.1.27), (5.1.28), (5.1.28)-Calib. and (5.1.26)	165
5.3	Iteration Number vs. $\ x - y\ _2$ for Method 5.2.2.2	182
5.4	Turbined volumes (m^3) for each reservoir of the Ain Valley. From top left to bottom right, solutions produced by Reduced Lagrangian Dual (§5.2.2.1.1), Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2), Reduced Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2.2), Reduced Augmented Lagrangian with Bundle Updating and Reduced Augmented Lagrangian + DQA (§5.2.2.3)	185
5.5	Turbined volumes (m^3) for each reservoir of the Isere Valley. From top left to bottom right, solutions produced by Reduced Lagrangian Dual (§5.2.2.1.1), Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2), Reduced Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2.2), Reduced Augmented Lagrangian with Bundle Updating and Reduced Augmented Lagrangian + DQA (§5.2.2.3)	186

5.6	Thermal Generation (MW) for each thermal unit. From top left to bottom right, solutions produced by Reduced Lagrangian Dual (§5.2.2.1.1), Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2), Reduced Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2.2), Reduced Augmented Lagrangian with Bundle Updating and Reduced Augmented Lagrangian + DQA (§5.2.2.3)	187
5.7	Deviations from Load. 100 Scenarios for η in eq.(5.2.22) and lower/upper bounds derived from each of the solutions. Reduced Lagrangian Dual (RLD), Augmented Lagrangian + Auxiliary Problem principle (AL+APP), Reduced Augmented Lagrangian + Auxiliary Problem principle (RAL+APP), Reduced Augmented Lagrangian with Bundle Updating (RAL+APP+Bdl) and Reduced Augmented Lagrangian + DQA (RAL+DQA).	188

List of Tables

4.1	Comparison of Algorithms (nne^x stands for $nn10^x$), assuming a Slater Point available.	110
4.2	Comparison of Algorithms (nne^x stands for $nn10^x$), infeasible starting point. Precision of Genz' code $\varepsilon^g = 1e^{-4}$	111
4.3	Effect of "Genz Precision" (nne^x stands for $nn10^x$), Alg.PB.SEV with $K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0\kappa = 0.5$	111
4.4	Effect of noise test (nne^x stands for $nn10^x$), Alg.PB.SEV with $K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0\kappa = 0.5$, TOL = 0.5, and Genz precision $5e^{-4}$. . .	111
4.5	Effect of Descent test & Stopping Criteria (αe^x stands for $\alpha 10^x$). Alg.PB.SEV with parameter setting $K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0\kappa = 0.5$. Precision of Genz code is taken to be $5e^{-4}, \rho^k = 0, \alpha = 0, \beta = -1 + \varepsilon^m$	112
4.6	Effect of updating ρ^k and the noise test (αe^x stands for $\alpha 10^x$). Alg.PB.SEV with parameter setting $K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0\kappa = 0.5$, TOL = 0.5. Precision of Genz code is taken to be $5e^{-4}$	113
4.7	Comparison of Algorithms (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 5e^{-4}$. The number of iterations at which (4.2.10) is found infeasible is indicated in between brackets.	140
4.8	Effect of γ (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 5e^{-4}$. The number of iterations at which (4.2.10) is found infeasible is indicated in between brackets.	141
4.9	Effect of adding (4.2.17) to Step 5. (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 5e^{-4}$	141
4.10	Effect of precise Oracle. (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 1e^{-4}$	142
4.11	Comparison of Alg.PB and Alg.LB on the Isère Valley. (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 5e^{-4}$. The number of iterations at which (4.2.10) is found infeasible is indicated in between brackets. The indication $[\neg(4.2.17)]$ means that auxiliary problem (4.2.17) is not solved.	144
4.12	Effect of precise Oracle. (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 1e^{-4}$	145

5.1	Comparison of costs and number of violations	163
5.2	Comparison of costs and number of violations	163
5.3	Thermal Plant data	180
5.4	Results for the Lift and Dualize Method	181
5.5	Results for the Lift and Dualize - Augmented Lagrangian Methods (The objective function value is shown if a feasible solution is generated)	182
5.6	Results for the Lift and Dualize - Augmented Lagrangian Methods - Reduced (The objective function value is shown if a feasible solution is generated)	183
5.7	Results for the Lift and Dualize - Augmented Lagrangian Methods - Reduced - Updating with Bundle Method (The objective function value is shown if a feasible solution is generated)	184
5.8	Results for the Diagonal Quadratic Approximation (The objective function value is shown if a feasible solution is generated)	184

Chapter 1

Introduction

1.1 Introduction

Energy management optimization problems deal with decision making problems with time spans ranging from intra-daily to several decades. Problems are as varied as:

1. computing day-ahead, the production schedule in a hydro-thermal generation system
2. obtaining a strategy of use of water and opportunity costs for the coming year
3. computing the maintenance schedule of thermal plants
4. obtaining investment decisions

It is clear that uncertainty intervenes in each of these problems. For instance in the first problem, the schedule is determined before observing uncertainty. Random deviations of load, renewable generation and (partial) outages then lead to offer-demand mismatches. These have to be dealt with appropriately. For instance, by setting spinning reserve requirements. Uncertainty on inflows also impacts the production schedule, but may render it infeasible. Indeed, for a fixed production schedule, excessive inflows may bring forth violations of the upper reservoir bound. Similarly, a lack of inflows may bring forth violations of the lower bound. As a matter of fact the earlier mentioned uncertainty factors also make the production schedule infeasible since the offer-demand balance is no longer satisfied. In the second and third problem, the decision taken in a specific time stage of the problem is assumed to have knowledge of all uncertainty that has preceded (dynamic decision making). In this aspect it differs from the first and fourth problem wherein the decisions are taken prior to observing uncertainty (static decision making). Dynamic decision making problems are frequently formulated as (Stochastic) Dynamic Programming problems or as optimization problems on a scenario tree (e.g., [119, 95, 91]). This essentially renders the problem deterministic (but large scale). Formulating constraints involving both the decision vector and a random variable is then nearly as

easy as in the deterministic setting. This becomes more complicated when we assume that decisions are taken prior to observing uncertainty. For example, what is meant with x being feasible in a linear inequality system $Ax \leq b$ when both A and b are random variables? Yet, such a question has to be answered if we are to solve problem 1 highlighted above. Two fields of optimization provide a framework for giving a sensible meaning to such a "random inequality system". These are Robust Optimization ([13]) and Probabilistic Programming ([181]). We shall (nearly) exclusively focus on the latter approach. As an application we will investigate to what extent a production schedule can be computed while integrating uncertainty on both inflows and load.

Probabilistic constraints are encountered in many engineering problems involving uncertain data. We can find applications in water management, telecommunications, electricity network expansion, mineral blending, chemical engineering etc. (e.g., [105, 155, 250, 185, 187, 246, 239]) For an overview of theory, numerics and applications of chance constraints we refer to [51, 181, 182] and references therein.

Initiated by Charnes and Cooper [32] and pioneered by Prékopa (e.g., by his celebrated log-concavity-Theorem [178]) the analysis of probabilistic constraints has attracted much attention in recent years with a focus on algorithmic approaches. Without providing an exhaustive list, we refer here to models like robust optimization [14], penalty approach [69], p-efficient points [53, 54], scenario approximation [29], sample average approximation [168] or convex approximation [160].

In this Thesis we will investigate probabilistic constraints following the traditional (as initiated by Prékopa) approach. We will investigate structural properties such as differentiability and convexity. We will also provide two specially designed Algorithms for dealing with (convex) optimization problems under probabilistic constraints. Globally the work can be seen as a "quest for tractability". Unlike commonly stated, we firmly believe that probabilistic programming is perfectly tractable in most/many situations of interest. To illustrate some advances it is of interest to recall that J. Mayer (in [149]) stated that the largest probabilistic program solved before 2000 and published had a random vector in dimension 7. In the same work a problem in dimension 30 is solved, requiring up to 800 minutes of CPU time. In this Thesis, we illustrate one of the algorithms on an instance having a random vector in dimension 168. Computation time is "only" around 350 minutes. It is likely that such large scale probabilistically constrained problems only allow for feasible solutions because of a strong underlying correlation structure. It might also be required that some constraints are inactive in a to be defined probabilistic sense.

This document is organized around four chapters. Chapter 2 gives an introduction to the known theory and structural properties of probabilistic constraints. These properties will be used frequently in the document. This chapter also helps as a guide to show how newly derived results fit in. Extensions of theory and algorithms can be found in Chapters 3 and 4 respectively. Chapter 5 is devoted to applications in energy management. In particular, we investigate cascaded reservoir management, robust against uncertainty on inflows and unit-commitment, robust against uncertainty on both inflows and load.

1.2 Main Contribution of this Work

In section 3.1 we investigate to what extent a characterization of the feasible set of a chance constraint can be derived involving simpler chance constraints. An example shows the interest of the suggested approach. We then provide several sufficient (but not necessary) conditions with negative results. These results hold in particular for the specially structured probabilistic constraints appearing in the applications of Chapter 5. To the best of our knowledge these ideas have not been investigated yet.

Many algorithms of non-linear programming require the knowledge of (sub-)gradients of the constraints in order to numerically solve a problem. If these constraints are probabilistic constraints, we need to be able to evaluate such a gradient as efficiently as possible. Very general differentiability statements exist ([237]) that represent the gradient of a probabilistic constraint as the sum of an integral over a volume and integral over a surface. Numerical use of these formulae appears to be quite difficult. Many other approaches investigate special cases. These results are of interest because they link a component of the partial derivative of a probabilistic constraint and the evaluation of another probabilistic constraint. Frequently the latter constraint is of similar nature as the one we took the derivative from. This means that if one is able to evaluate a probabilistic constraint (clearly a prerequisite), one can also compute its derivative. These differentiability results exist for the derivative of non-degenerate Gaussian distribution functions [175], multi-variate Gamma distribution functions [186] and multi-variate Dirichlet distributions [221]. The results of section 3.2 (published in [244, 246]) provide efficient gradient (and Hessian) formulae of the above type for specially structured probabilistic constraint appearing in Energy Management Applications. These results extend [175] to bilateral constraints involving a non-degenerate Gaussian random variable. We also provide a result ([247]) involving the computation of the gradient of a probabilistic constraint with "Gaussian Matrix Uncertainty". The results appearing in section 3.2 were derived with R. Henrion, A. Möller and R. Zargati.

Differentiability of probabilistic constraints is further examined in Section 3.3 where efficient gradient formulae are derived for probabilistic constraints involving a Gaussian (or Student) random variable linked to the decision vector through a rather general mapping. Special cases involve log-normal and chi-squared random variables. These results are taken from a draft submitted with R. Henrion ([242]).

Convexity of the feasible set is an important feature in probabilistic programming. Occasionally such convexity depends on the requested safety level. This is known as eventual convexity and clearly suffices in practical applications. Early results on eventual convexity were derived for specific individual probabilistic constraints [121]. Recently [110, 111] provided new insights in eventual convexity of probabilistically constrained feasible sets. The involved multi-variate random variable has components "correlated" through a specially structured Copulae. The results ([240]) in Section 3.4 provide an extension of the latter results on eventual convexity, by allowing for more Copulae and providing lower thresholds. Such lower thresholds allow us to prove convexity for probabilistically constrained feasible set with lower safety-levels.

In sections 4.1 and 4.2 we provide two Bundle-Algorithms for dealing with convex constrained optimization problems. The constraint is an appropriate transform of the probabilistic constraint making it convex. We thus (implicitly) assume that the elements appearing in the constraint satisfy appropriate hypothesis for this to hold. We also assume that the value of the constraint and a (sub-)gradient can be computed. It is not reasonable to assume that this can be done up to arbitrary precision. The suggested algorithms are designed in such a way that this imprecision is dealt with. The specific case of probabilistic constraints led to the definition of an "upper"-oracle and considerably complicates convergence analysis. The results in Section 4.1 are taken from a joint work [248] with C. Sagastizábal, whereas the results of Section 4.2 are taken from a joint work [241] with W. de Oliveira.

Chapter 5 contains applications involving some of the largest problems featuring probabilistic constraints seen in the literature so far ([149]). Especially the global unit-commitment problem of Section 5.2 requires iteratively solving many probabilistic programs. It is intuitively felt that setting up a robust unit-commitment problem with a probabilistic constraint on the offer-demand equilibrium leads to a tractable model. Several decomposition approaches are suggested and experimented with on a typical example. Making cascaded reservoir management robust against uncertainty on inflows is investigated in Section 5.1. Those results are partially taken from [246], which is a joint work with R. Henrion, A. Möller and R. Zorgati. Most papers on cascaded reservoir management with chance constraints consider the easier case of individual chance constraints (e.g., [143, 63, 65, 142, 155]). Although we should consider joint chance constraints from a robustness perspective, this only rarely done [185, 186]. Also most of these papers only consider very simple models without serially linked reservoirs, flow delay, time series modelling of statistical data and have random vectors in small dimension.

Chapter 2

Preliminaries

Chance constrained programming is the branch of Stochastic Programming dealing with constraints of the form

$$\mathbb{P}[g(x, \xi) \geq 0] \geq p, \quad (2.0.1)$$

where $x \in \mathbb{R}^n$ is the decision vector, $\xi \in \mathbb{R}^m$ a random variable and $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^k$ a constraint mapping. The level $p \in (0, 1)$ is user given and defines the preference for safety of the decision x . The constraint (2.0.1) means that we wish to take a decision x that satisfies the k -dimensional random inequality system $g(x, \xi) \geq 0$ with high enough probability. Such chance constraints naturally arise in engineering problems when in usual constraints $h(x) \geq 0$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^k$ one identifies unknown parameters or random vectors. The typical situation is one wherein $h(x) := g(x, \mathbb{E}(\xi))$, e.g., $\mathbb{E}(\xi)$ can be identified with a forecast of ξ . Constraints of the form (2.0.1) can then be formed upon realizing that the forecast was not accurate enough or additional engineering securities had to be built into h in order to account for variations in ξ .

Now two situations may arise, one wherein we consider that the distribution of ξ is unknown, insufficiently characterized or simply we forbid ourselves to use it. In such a situation constraints of the form (2.0.1) can be replaced by constraints of the form

$$g(x, z) \geq 0, \quad \forall z \in \Theta, \quad (2.0.2)$$

where Θ is an appropriately chosen, explicitly defined set called uncertainty set. This is a version of "Robust Optimization". We refer to the book [13] and references therein for a full treatment of this topic. In the second situation we assume that ξ is well-characterized and that knowledge of the distribution is available. Then constraints of the form (2.0.1) can be investigated and this is the hypothesis that we make in this Thesis. Such a situation naturally arises when ξ has been investigated by Statisticians or Econometrists. Such is frequently the situation when in decision making problems sensitivity or risk analyses are conducted. Often such analyses are based on applying an optimization tool on a set of generated scenarios for ξ . The scenario generator is frequently based on a time-series decomposition of ξ , at least when a temporal effect is clearly identified or otherwise on a finely characterized distribution of ξ .

Two conceptually different versions of (2.0.1) exist and are referred to as Individual Chance Constraints (ICC) or Joint Chance Constraints (JCC). The equation appearing in (2.0.1) is a version of a Joint Chance Constraint. A deduced set of Individual Chance Constraints would be

$$\mathbb{P}[g_i(x, \xi) \geq 0] \geq p_i, i = 1, \dots, k, \quad (2.0.3)$$

where g_i refers to the i -th component of the mapping g and $p_i \geq p$ are arbitrary choices. The situation of (2.0.3) refers to a situation wherein we wish to satisfy each individual equation in the random k -dimensional inequality system $g(x, \xi) \geq 0$ with high enough probability, but we make no request on the system as a whole. This loss of robustness is compensated, in many situations, by an easier numerical and theoretical treatment. From an engineering perspective one can also argue that (2.0.3) offers already increased robustness with respect to solving deterministic problems wherein ξ was replaced with a forecast, e.g., $\mathbb{E}(\xi)$. One readily observes that a feasible solution for (2.0.1) is feasible for (2.0.3) and that the inverse holds whenever $\sum_{i=1}^k p_i \geq p + (k - 1)$. Despite this inner and outer approximation of the feasible set of (2.0.1), we believe that it is worthwhile to consider the JCC of (2.0.1). This is exactly what we will look at in this Thesis.

An important special case of (2.0.1) is one wherein x and ξ are not coupled through the mapping g but appear separate. The mapping g is then of the form $g(x, \xi) = h(x) - \tilde{g}(\xi)$. The constraint (2.0.1) then becomes:

$$\mathbb{P}[h(x) - \tilde{g}(\xi) \geq 0] \geq p \quad (2.0.4)$$

and is referred to as a separable (joint) chance constraint.

Chance constraints were first formulated as ICCs by [35] and further developed in [32, 33, 34]. Joint Chance Constraints were first formulated by [152]. The field has then received major contributions by Prékopa in the early 70s [175, 178] with in particular very general convexity results for the feasible set of (2.0.1). Other key contributions are from Szántai (e.g., [186, 221, 222]), Uryasev (e.g., [235]), Henrion (e.g., [97, 99, 108, 106]), Römisch (e.g., [109, 204]), Dentcheva and Ruszczyński (e.g., [54]).

In this chapter we will sketch an overview of known results on structural properties of probabilistic constraints of the form (2.0.1). It will serve as a guideline to show where our results fit in.

2.1 Measurability

Even though one easily writes a constraint of the type (2.0.1) a first question consists of its well-foundedness. In particular, for a fixed $x \in \mathbb{R}^n$ is the set $\{\omega : g(x, \xi(\omega)) \geq 0\}$ measurable so that $\mathbb{P}[g(x, \xi) \geq 0]$ is a meaningful composition ?

This can be answered positively in quite a general setting. For this analysis the dependence on x will be neglected. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Consider a random vector $\xi : \Omega \rightarrow E$, where $(E, \mathcal{B}(E))$ is a topological vector space equipped with its Borel-sigma algebra $\mathcal{B}(E)$. Now consider a mapping $g : E \rightarrow \mathbb{R}^k$. It is then clear that Borel

measurability of g suffices for the set

$$\{\omega \in \Omega : g(\xi(\omega)) \geq 0\},$$

to be \mathcal{F} -measurable. We can thus write

$$\mathbb{P}[g(\xi) \geq 0],$$

as this is now a well defined expression. In particular every upper-semi-continuous mapping g is Borel-Measurable.

This obvious statement hides some difficulties. In particular, it is not sufficient for g to be concave, since not all convex sets are Borel measurable. If ξ admits a density with respect to the Lebesgue measure, the above difficulty disappears since convex sets are Lebesgue measurable ([133, 67]).

We can thus conclude that as long as $\xi \in \mathbb{R}^m$ appearing in (2.0.1) admits a density and $z \mapsto g(x, z)$ is concave (or upper-semi-continuous), the constraint (2.0.1) is well-defined. The most common forms of inequality systems $g(x, \xi) \geq 0$ are those wherein g represents a linear random inequality system of the form $Ax \geq b$. Either components of A or b are elements of the vector ξ . Measurability in this case is thus assured.

2.2 Closedness of the Feasible Set

A second question of importance is whether the feasible set

$$M(p) := \{x \in \mathbb{R}^n : \mathbb{P}[g(x, \xi) \geq 0] \geq p\} \tag{2.2.1}$$

is actually closed. This is of-course a necessary property for any optimization problem with a constraint of the type (2.0.1) to actually attain an optimum. This can be shown under fairly general assumptions on the mapping g . In what follows we (trivially) generalize the known results to metric spaces and an arbitrary set of constraints. In this we follow the work [100], but the original results are published in [190, 191, 192]. We can also cite [203] for a result similar to Lemma 2.2.2 below.

Lemma 2.2.1. *Let (E, d) be a metric space, with metric $d : E \times E \rightarrow \mathbb{R}_+$. Consider an arbitrary mapping $h : E \rightarrow \mathbb{R}$. The level sets $L_t = \{x \in E : h(x) \geq t\}$ are closed for all $t \in \mathbb{R}$ if and only if h is upper semi-continuous.*

Proof. Fix an arbitrary t and pick any sequence $x_n \in L_t$, converging to limit $x \in E$. Since E is a metric space, upper semi continuity of h gives

$$h(x) \geq \limsup_{y \rightarrow x} h(y) = \limsup_{n \rightarrow \infty} h(x_n) \geq t,$$

so $x \in L_t$ and L_t is therefore closed. Conversely let $x \in E$ be arbitrary and pick a sequence $x_n \rightarrow x$ such that

$$t := h(x) = \limsup_{y \rightarrow x} h(y) = \lim_{n \rightarrow \infty} h(x_n),$$

Then for arbitrary $\varepsilon > 0$, there exists some N with $h(x_n) \geq t - \varepsilon$ for all $n \geq N$, so $x_n \in L_{t-\varepsilon}$ for those n . From closedness of L_t for all t this gives $x \in L_{t-\varepsilon}$, therefore

$$h(x) \geq \limsup_{y \rightarrow x} h(y) - \varepsilon.$$

since $\varepsilon > 0$ is arbitrary, h is upper semi continuous. \square

Lemma 2.2.2. *Let E, F be arbitrary metric spaces, T be an arbitrary index set and let $\xi \in F$ be a random variable. Assume furthermore that all members $g_t : E \times F \rightarrow \mathbb{R}$ of the family of mappings $\{g_t\}_{t \in T}$ are (jointly) upper semi continuous. The mapping $\varphi : E \rightarrow [0, 1]$ defined as $\varphi(x) = \mathbb{P}[g_t(x, \xi) \geq 0 \forall t \in T]$ is then upper semi-continuous. As a consequence, the set $M(p) := \{x : \varphi(x) \geq p\}$ is closed for all $p \in [0, 1]$.*

Proof. We begin by observing that $M(p) = \emptyset$ if $p > 1$ and $M(p) = E$ if $p < 0$ and that these sets are closed. The asserted upper semi-continuity of the mapping φ is therefore equivalent with $M(p)$ being closed as a consequence of Lemma 2.2.1.

Let $p \in [0, 1]$ be arbitrary. We will show that $M(p)$ is a closed set. To this end, pick an arbitrary converging sequence $x_n \in M(p)$ with limit point x , i.e., $\lim_{n \rightarrow \infty} x_n = x$, where convergence is in E . We will show that $x \in M(p)$.

To this end, define

$$\begin{aligned} H(x) &= \{z \in F : g_t(x, z) \geq 0 \forall t \in T\} \\ A_k &= \bigcup_{y : d_E(y, x) \leq k^{-1}} H(y), k \geq 1. \end{aligned}$$

Then $\mathbb{P}[\xi \in H(x)] = \mathbb{P}[g_t(x, \xi) \geq 0 \forall t \in T]$ and in particular $\mathbb{P}[\xi \in H(x_n)] \geq p$. We now claim that

$$H(x) = \bigcap_{k \geq 1} A_k. \quad (2.2.2)$$

Clearly $z \in H(x)$ implies $z \in A_k$ for all $k \geq 1$. To show the opposite, let $z \in \bigcap_{k \geq 1} A_k$, be arbitrary. Then one can find sequences $z_k \rightarrow z$, $y_k \rightarrow x$, such that $z_k \in H(y_k)$ and from upper semi-continuity of g_t we get for each $t \in T$:

$$g_t(x, z) \geq \limsup_{k \rightarrow \infty} g_t(y_k, z_k) \geq 0, \quad (2.2.3)$$

yielding $z \in H(x)$. We have thus shown (2.2.2). Since clearly $A_{k+1} \subseteq A_k$ for all $k \geq 1$ it follows from standard measure theory [19] that $\mathbb{P}[\xi \in A_k] \downarrow \mathbb{P}[\xi \in H(x)]$. Now pick an arbitrary $\varepsilon > 0$ and $k' \geq 0$ such that $\mathbb{P}[\xi \in H(x)] - \mathbb{P}[\xi \in A_{k'}] \geq -\varepsilon$. Such k' can be found according to the statements above. Since x_n converges to x in E , one can moreover find N such $d_E(x_n, x) \leq k'^{-1}$ for all $n \geq N$. According to the definition of $A_{k'}$ this implies that $H(x_n) \subseteq A'_{k'}$ for $n \geq N$. Altogether we obtain

$$\mathbb{P}[\xi \in H(x)] - \mathbb{P}[\xi \in H(x_n)] \geq \mathbb{P}[\xi \in H(x)] - \mathbb{P}[\xi \in A_{k'}] \geq -\varepsilon, \quad (2.2.4)$$

i.e., $\mathbb{P}[\xi \in H(x)] \geq p - \varepsilon$. Since $\varepsilon > 0$ is arbitrary, $\mathbb{P}[\xi \in H(x)] \geq p$ and $x \in M(p)$ as was to be shown. \square

2.3 Continuity of Probabilistic Constraints

In order to show continuity of the probabilistic constraint we require a situation wherein no probabilistic mass is assigned to the boundary of the set $\{z \in F : g(x, z) = 0\}$, where the situation is as in Lemma 2.2.2. An example of a discontinuous probability constraint can be found in [100, Beispiel 2.1]. The following result is a straightforward extension of [100, 190].

Lemma 2.3.1. *Let E, F be arbitrary metric spaces, T be an arbitrary index set and let $\xi \in F$ be a random variable. Assume furthermore that all members $g_t : E \times F \rightarrow \mathbb{R}$ of the family of mappings $\{g_t\}_{t \in T}$ are (jointly) lower semi-continuous. Assume furthermore that the sets $N_x := \{z \in F : \inf_{t \in T} g_t(x, z) = 0\}$ are \mathbb{P} -null-sets for all $x \in E$, i.e., $\mathbb{P}[\xi \in N_x] = 0$. The mapping $\varphi : E \rightarrow [0, 1]$ defined as $\varphi(x) = \mathbb{P}[g_t(x, \xi) \geq 0 \forall t \in T]$ is then also lower semi-continuous.*

Proof. Fix $\bar{x} \in E$ arbitrarily and define the mapping $g^{inf} : E \times F \rightarrow \mathbb{R}$ as $g^{inf}(x, z) := \inf_{t \in T} g_t(x, z)$. The mapping g^{inf} is then also lower semi-continuous and $-g^{inf}$ upper semi-continuous. Upon applying Lemma 2.2.2 we derive

$$\limsup_{x \rightarrow \bar{x}} \mathbb{P}[-g^{inf}(x, \xi) \geq 0] \leq \mathbb{P}[-g^{inf}(\bar{x}, \xi) \geq 0]. \quad (2.3.1)$$

According to our assumption we have $\mathbb{P}[g^{inf}(x, \xi) = 0] = 0$. So combining this with (2.3.1) yields:

$$\begin{aligned} \liminf_{x \rightarrow \bar{x}} \varphi(x) &= \liminf_{x \rightarrow \bar{x}} \mathbb{P}[g_t(x, \xi) \geq 0 \forall t \in T] = \liminf_{x \rightarrow \bar{x}} \mathbb{P}[g^{inf}(x, \xi) \geq 0] = \\ &= \liminf_{x \rightarrow \bar{x}} \mathbb{P}[g^{inf}(x, \xi) > 0] = - \limsup_{x \rightarrow \bar{x}} -\mathbb{P}[g^{inf}(x, \xi) > 0] = \\ &= - \limsup_{x \rightarrow \bar{x}} (\mathbb{P}[g^{inf}(x, \xi) \leq 0] - 1) = 1 - \limsup_{x \rightarrow \bar{x}} \mathbb{P}[g^{inf}(x, \xi) \leq 0] \\ &\geq 1 - \mathbb{P}[g^{inf}(\bar{x}, \xi) \leq 0] = \mathbb{P}[g^{inf}(\bar{x}, \xi) > 0] = \mathbb{P}[g^{inf}(\bar{x}, \xi) \geq 0] = \varphi(\bar{x}). \end{aligned}$$

We have thus shown that φ is indeed lower semi-continuous. \square

Remark 2.3.2. *If T is a finite set, it is sufficient to require that $N_x := \{z \in F : g_t(x, z) = 0\}$ is a \mathbb{P} -null set for each $t \in T$.*

Upon combining Lemma 2.3.1 and 2.2.2 we thus obtain the following Theorem providing conditions for continuity of the probabilistic constraint:

Theorem 2.3.3. *Let E, F be arbitrary metric spaces, T be an arbitrary index set and let $\xi \in F$ be a random variable. Let all members $g_t : E \times F \rightarrow \mathbb{R}$ of the family of mappings $\{g_t\}_{t \in T}$ be (jointly) continuous. Assume furthermore that the sets $N_x := \{z \in F : \inf_{t \in T} g_t(x, z) = 0\}$ are \mathbb{P} -null-sets for all $x \in E$, i.e., $\mathbb{P}[\xi \in N_x] = 0$. The mapping $\varphi : E \rightarrow [0, 1]$ defined as $\varphi(x) = \mathbb{P}[g_t(x, \xi) \geq 0 \forall t \in T]$ is then also continuous.*

An important specification is obtained when $F = \mathbb{R}^m$:

Lemma 2.3.4. *Let E be an arbitrary metric space, T be an arbitrary index set and let $\xi \in \mathbb{R}^m$ be a random variable having a density with respect to the Lebesgue measure. Let all members $g_t : E \times \mathbb{R}^m \rightarrow \mathbb{R}$ of the family of mappings $\{g_t\}_{t \in T}$ be (jointly) continuous. Assume furthermore that the sets $N_x := \{z \in \mathbb{R}^m : \inf_{t \in T} g_t(x, z) = 0\}$ are Lebesgue-null-sets for all $x \in E$. The mapping $\varphi : E \rightarrow [0, 1]$ defined as $\varphi(x) = \mathbb{P}[g_t(x, \xi) \geq 0 \forall t \in T]$ is then continuous.*

Proof. Since ξ admits a density with respect to the Lebesgue measure it is clear that $\mathbb{P}[\xi \in N_x] = 0$ for all $x \in E$. We can now use Theorem 2.3.3 to conclude the proof. \square

2.4 Connectedness of Feasible Set

One of the last most general results we can derive is that of connectedness of the feasible sets of constraints of the type (2.0.1).

Theorem 2.4.1. *Let E, F be arbitrary metric spaces, T be an arbitrary index set and let $\xi \in F$ be a random variable. Let all members $g_t : E \times F \rightarrow \mathbb{R}$ of the family of mappings $\{g_t\}_{t \in T}$ be (jointly) quasi-concave. Assume furthermore that g is max-stable, i.e., for each $x_1, x_2 \in E$, one can find $x_3 \in E$ such that $g_t(x_3, z) \geq \max\{g_t(x_1, z), g_t(x_2, z)\}$ for each $z \in F, t \in T$. The set $M(p) := \{x \in E : \varphi(x) \geq p\}$ is then path-connected for each $p \in [0, 1]$, where $\varphi : E \rightarrow [0, 1]$ is defined as $\varphi(x) = \mathbb{P}[g_t(x, \xi) \geq 0 \forall t \in T]$.*

Proof. This follows directly from [97, Theorem 2.1] upon inverting the inequalities figuring therein. \square

2.5 Convexity of the Feasible Set

For a (highly) efficient numerical treatment it is clearly of interest to know under which conditions convexity of the feasible set for constraints of the type (2.0.1) is to be expected. Upon examining the special case wherein $g(x, z) = x - z$, in which the constraint (2.0.1) directly relates to the distribution function of ξ , it is clear that concavity of $x \mapsto \mathbb{P}[g(x, \xi) \geq 0]$ can't be expected. Indeed, already one dimensional distribution functions are not concave. However it is sufficient that the mapping $x \mapsto \mathbb{P}[g(x, \xi) \geq 0]$ is quasi-concave for the feasible set $M(p) := \{x \in \mathbb{R}^n : \mathbb{P}[g(x, \xi) \geq 0] \geq p\}$ to be convex.

Lemma 2.5.1. *Consider the situation of (2.0.1), define $\varphi : \mathbb{R}^n \rightarrow [0, 1]$ as $\varphi(x) = \mathbb{P}[g(x, \xi) \geq 0] \geq p$ and let $M(p)$ be defined as $M(p) := \{x \in \mathbb{R}^n : \mathbb{P}[g(x, \xi) \geq 0] \geq p\}$. Then the feasible set $M(p)$ is convex for all $p \in [0, 1]$ if and only if φ is quasi-concave.*

Proof. Let $p \in [0, 1]$, $x, y \in M(p)$, $\lambda \in [0, 1]$ all be arbitrary and form $x^\lambda = \lambda x + (1 - \lambda)y$. Quasi-concavity of φ implies $\varphi(x^\lambda) \geq \min\{\varphi(x), \varphi(y)\} \geq p$, showing $x^\lambda \in M(p)$. To show the other implication let $x, y \in \mathbb{R}^n$, $\lambda \in [0, 1]$ all be arbitrary. Upon defining $p := \min\{\varphi(x), \varphi(y)\}$ it follows that $x, y \in M(p)$. Hence from convexity of $M(p)$ we obtain $x^\lambda \in M(p)$, i.e., $\varphi(x^\lambda) \geq p$ as was to be shown. \square

Quasi-concavity (and further generalized) concavity has a good chance of arising in many situations. To this end we introduce the mapping:

Definition 2.5.2. Let $\alpha \in [-\infty, \infty]$ and $m_\alpha : \mathbb{R}_+ \times \mathbb{R}_+ \times [0, 1] \rightarrow \mathbb{R}$ be defined as follows

$$m_\alpha(a, b, \lambda) = 0 \text{ if } ab = 0, \quad (2.5.1)$$

for $a > 0, b > 0, \lambda \in [0, 1]$:

$$m_\alpha(a, b, \lambda) = \begin{cases} a^\lambda b^{1-\lambda} & \text{if } \alpha = 0 \\ \max\{a, b\} & \text{if } \alpha = \infty \\ \min\{a, b\} & \text{if } \alpha = -\infty \\ (\lambda a^\alpha + (1 - \lambda)b^\alpha)^{\frac{1}{\alpha}} & \text{else} \end{cases} \quad (2.5.2)$$

We can now provide the definition of generalized concavity:

Definition 2.5.3. A non-negative function f defined on some convex set $C \subseteq \mathbb{R}^n$ is called α -concave ($\alpha \in [-\infty, \infty]$) if and only if for all $x, y \in C, \lambda \in [0, 1]$:

$$f(\lambda x + (1 - \lambda)y) \geq m_\alpha(f(x), f(y), \lambda), \quad (2.5.3)$$

where m_α is as in Definition 2.5.2.

Remark 2.5.4. A function f is 0-concave if its logarithm is concave. For $\alpha \neq 0, \alpha \in \mathbb{R}$, the function f is α -concave if either f^α is concave for $\alpha > 0$ or f^α is convex for $\alpha < 0$. Quasi-concavity refers to $-\infty$ -concavity in Definition 2.5.3.

The following lemma can be found in [51] and shows that α -concavity implies quasi-concavity, i.e., quasi-concavity is the weakest form of "generalized" concavity that exists. In particular concave mappings are log-concave etc...

Lemma 2.5.5. Let m_α be the mapping as defined in Definition 2.5.2. The mapping $\alpha \mapsto m_\alpha$ is nondecreasing and continuous.

For calculus rules with α -concavity we refer to Theorems 4.19-4.23 of [51].

Definition 2.5.6. A probability measure \mathbb{P} defined on the Lebesgue measurable subsets of a convex set $C \subseteq E$ is said to be α -concave if for all Borel measurable subsets $A, B \subseteq C, \lambda \in [0, 1]$ we have

$$\mathbb{P}[\lambda A + (1 - \lambda)B] \geq m_\alpha(\mathbb{P}[A], \mathbb{P}[B], \lambda), \quad (2.5.4)$$

where $\lambda A + (1 - \lambda)B = \{\lambda x + (1 - \lambda)y : x \in A, y \in B\}$ is the Minkowski sum of A and B .

Remark 2.5.7. Prékopa introduced the notion of log-concave measures in [176] but the above generalization was suggested by [21, 22].

In fact it turns out that quasi-concavity of measures relates to densities. In fact "all" non-degenerate quasi-concave measures must have a density:

Lemma 2.5.8 ([21]). *If \mathbb{P} is a quasi-concave measure on \mathbb{R}^m and the dimension of its support is also m , then \mathbb{P} has a density with respect to the Lebesgue measure.*

Now that we know that quasi-concave measures have densities, it is natural to wonder if generalized concavity of such densities carries over to the measure. This turns out to be the case as the following result shows. We also refer to [181, Chapter 4] and references therein.

Theorem 2.5.9 ([176, 178, 22, 196]). *Let C be a convex subset of \mathbb{R}^m and let $s > 0$ be the dimension of the smallest affine subspace L containing C . The probability measure \mathbb{P} on C is γ -concave with $\gamma \in [-\infty, \frac{1}{s}]$ if and only if its probability density function with respect to the Lebesgue measure on L is α -concave with*

$$\alpha = \begin{cases} \frac{\gamma}{1-s\gamma} & \text{if } \gamma \in (-\infty, \frac{1}{s}) \\ -\frac{1}{s} & \text{if } \gamma = -\infty \\ \infty & \text{if } \gamma = \frac{1}{s} \end{cases} \quad (2.5.5)$$

Corollary 2.5.10. *Let $f : \mathbb{R}^m \rightarrow \mathbb{R}$ be integrable and positive on a convex set $C \subseteq \mathbb{R}^m$ with non-zero Lebesgue Measure. If f is α -concave on C with $\alpha \in [-\frac{1}{m}, \infty]$ and positive on the interior of C , then the measure $\mathbb{P}[A] = \frac{1}{\int_C f d\lambda} \int_A f d\lambda$ is γ -concave on C with*

$$\gamma = \begin{cases} \frac{\alpha}{1+m\alpha} & \text{if } \alpha \in (\frac{1}{m}, \infty) \\ \frac{1}{m} & \text{if } \alpha = \infty \\ -\infty & \text{if } \alpha = -\frac{1}{m} \end{cases} \quad (2.5.6)$$

Lemma 2.5.11. *Let $\xi \in \mathbb{R}^m$ be a random variable with α -concave probability distribution \mathbb{P}_ξ , where $\alpha \in [-\infty, \infty]$. Let T be an $s \times m$ constant matrix and let $\eta \in \mathbb{R}^s$ be the random variable defined as $\eta := T\xi$. Then the probability distribution of η is α -concave too.*

Remark 2.5.12. *For the application of Theorem 2.5.9 it is important to distinguish probability distribution and probability distribution function. The former concept relates to the measure \mathbb{P} defined on an appropriate σ -algebra. The latter concept is a restriction of the measure to sets of the form $(-\infty, x]$. In order to illustrate the difference, the log-normal law has a probability distribution without any generalized concavity property, but probability distribution function that is log-concave.*

In applications the following theorem (proved for multivariate normal distributions in [261]) provides a strong tool for deriving convexity of feasible sets:

Theorem 2.5.13 ([176]). *Let $C \subseteq \mathbb{R}^m$ be a convex set and let \mathbb{P} be a α -concave measure defined on Lebesgue measurable subsets of C , where $\alpha \in [-\infty, \infty]$. Let A be a convex (i.e., Lebesgue measurable) subset of C and $x \in \mathbb{R}^m$ arbitrary, then $x \mapsto f(x) := \mathbb{P}[A+x]$ is an α -concave function.*

Proof. Let $x, y \in \mathbb{R}^m, \lambda \in [0, 1]$ all be arbitrary and remark that convexity of A provides the following invariance under the Minkowski sum $A = \lambda A + (1 - \lambda)A$. It thus follows

that $\lambda(A + x) + (1 - \lambda)(A + y) = A + \lambda x + (1 - \lambda)y$. Upon applying the α -concavity of the measure \mathbb{P} we obtain:

$$\begin{aligned} f(\lambda x + (1 - \lambda)y) &= \mathbb{P}[A + \lambda x + (1 - \lambda)y] = \mathbb{P}[\lambda(A + x) + (1 - \lambda)(A + y)] \\ &\geq m_\alpha(f(x), f(y), \lambda). \end{aligned}$$

□

Theorem 2.5.14 ([176]). *Let $\xi \in \mathbb{R}^m$ be a random variable that induces an α -concave probability distribution \mathbb{P} , where $\alpha \in [-\infty, \infty]$. Then its associated probability distribution function is also α -concave.*

Proof. Define the convex set $A = \{z \in \mathbb{R}^m : z \leq 0\}$. Then $x, y \in \mathbb{R}^m$ with $y \leq x \Leftrightarrow y \in A + x$. As a consequence $F_\xi(x) := \mathbb{P}[\xi \leq x] = \mathbb{P}[\xi \in (A + x)]$. The result now follows from Theorem 2.5.13. □

One of the most general concavity results is the following. It can be found in this form in [51], but its original form is found in [177, 178] where it is specified to log-concave (i.e., 0-concave) measures. Tamm [227] observed that quasi-concavity of the mapping g was sufficient and concavity of g an excessive requirement. The generalization to α -concave measures was provided by [21, 22].

Theorem 2.5.15. *Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^k$ be a (jointly) quasi-concave function and let $\xi \in \mathbb{R}^m$ be a random variable inducing an α -concave probability distribution \mathbb{P} . Then the mapping $x \in \mathbb{R}^n \mapsto G(x) := \mathbb{P}[g(x, \xi) \geq 0]$ is an α -concave function on the set $D = \{x \in \mathbb{R}^n : \exists z \in \mathbb{R}^m \text{ with } g(x, z) \geq 0\}$.*

Remark 2.5.16. *Consider separable constraints in (2.0.1), i.e., $g_i(x, z) = h_i(x) - z$, $i = 1, \dots, k$ where $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Then $G(x) := \mathbb{P}[g(x, \xi) \geq 0] = \mathbb{P}[\xi \leq h(x)] = F_\xi(h(x))$ and convexity of the feasible set directly relates to the generalized concavity properties of distribution function and (generalized) concavity properties of h .*

Remark 2.5.17. *Consider an individual separable constraint of the form 2.0.3 (i.e., set $k = 1$ in (2.0.1)). One then easily derives that $\mathbb{P}[\xi \leq h(x)] \geq p$ is equivalent with $h(x) \geq F^{(-1)}(p)$, where the latter is the generalized inverse of the one-dimensional distribution function F . This is an explicit constraint and convexity can be deduced from properties of h directly.*

Remark 2.5.18. *The strength of Theorems 2.5.9 and 2.5.15 is that many random variables actually have generalized concavity properties. Indeed we can provide the following list of random variables having α -concave densities (see [182]):*

1. *Gaussian random variables have log-concave densities*
2. *Dirichlet distributions with parameters $\vartheta_j \geq 1$ are log-concave, whereas for $\vartheta_j \leq 1$, they are log-convex.*
3. *Uniform distributions are log-concave*

4. Wishart distributions (on the set of square matrices) are log-concave
5. Beta distributions (on the set of square matrices) are log-concave
6. Multi-variate Gamma distributions are log-concave
7. The m -variate Cauchy distributions has an $-\frac{1}{m}$ -concave density. Its distribution is therefore quasi-concave.
8. The Pareto distribution has a $-\frac{1}{m}$ -concave density. Its distribution is therefore quasi-concave.
9. The multi-variate Student t -distribution has a $-\frac{1}{m}$ -concave density. Its distribution is therefore quasi-concave.

2.5.1 Special Cases with Convexity

Joint quasi-concavity of the mapping g appearing in (2.0.1) does not hold in all cases of interest. This does not imply that convexity of the feasible set can't be derived. A particular case of interest is one wherein $g(x, \xi) \geq 0$ represents a random linear inequality system $Ax \leq b$, where A is random.

The following easily derived result provides a convexity statement in such a setting

Lemma 2.5.19 ([121]). *Consider the constraint of the form (2.0.1) where $k = 1$, $g(x, z) = z^\top x - b$ and $\xi \in \mathbb{R}^m$ is a multivariate Gaussian random variable. Then the feasible set $M(p)$ is convex for all $p > \frac{1}{2}$.*

Generalizations of this result to $k > 1$ require special correlation structures for the Gaussian random vector. The column version is due to [179], whereas the row version comes from [26]:

Theorem 2.5.20. *Let A be a $m \times n$ matrix having a non degenerate multivariate Gaussian density in \mathbb{R}^{nm} . Assume that there is a fixed covariance matrix C and a $n \times n$ (or $m \times m$) symmetric matrix S , such that either:*

1. The columns $A_{.j}$ of A , satisfy

$$\text{Cov}(A_{.j}, A_{.k}) = s_{jk}C, j, k = 1, \dots, n,$$

2. The rows $A_{j.}$ of A , satisfy

$$\text{Cov}(A_{j.}, A_{k.}) = s_{jk}C, j, k = 1, \dots, m,$$

Then the set $\{x : \mathbb{P}[Ax \leq 0] \geq p\}$ is convex for all $p \geq \frac{1}{2}$.

Proof. A very comprehensive proof can be found in [119, Theorem 2.12]. □

2.5.2 Eventual Convexity of the Feasible set

As in the result by Kataoka ([121]), convexity of the feasible set $M(p)$ cannot always be obtained for any probability level p . From a practical perspective this is not necessarily a problem since we are naturally looking for high p levels when formulating constraints of the type (2.0.1). In some cases, we can show that the feasible set is convex if p is large enough. Convexity of feasible sets with high enough p is known as eventual convexity. Some key results on eventual convexity have been derived by [110, 111]. The following concept plays a key role in their results.

Definition 2.5.21. *The function $f : \mathbb{R} \rightarrow \mathbb{R}$ is called r -decreasing ($r \in \mathbb{R}$) if it is continuous on $(0, \infty)$ and there exist some $t^* > 0$ such that $t \mapsto t^r f(t)$ decreases strictly for $t > t^*$.*

Lemma 2.5.22 ([110]). *Let $F : \mathbb{R} \rightarrow [0, 1]$ be a probability distribution function with a $\gamma + 1$ -decreasing density for some $\gamma > 0$. Then the function $z \mapsto F(z^{-\frac{1}{\gamma}})$ is concave on $(0, (t^*)^{-\gamma})$, with t^* as in Definition 2.5.21. Moreover $F(t) < 1$ for all $t \in \mathbb{R}$.*

Definition 2.5.23. *Let $q \in (0, 1)^m$ be arbitrary. A copula $C : [0, 1]^m \rightarrow [0, 1]$ is called log-exp concave on $[q, 1]$ if the mapping $u \in \mathbb{R}^m \mapsto \log C(e^{u_1}, \dots, e^{u_m})$ is concave on $[\log(q), 0]$.*

Theorem 2.5.24 ([110, 111]). *Let $\xi \in \mathbb{R}^m$ be a random vector, $C : [0, 1]^m \rightarrow [0, 1]$ be a copula, and let $g_i : \mathbb{R} \rightarrow \mathbb{R}$ be mappings such that*

$$\mathbb{P}[\xi \leq g(x)] = C(F_1(g_1(x)), \dots, F_m(g_m(x))), \quad (2.5.7)$$

where F_i are the marginal distribution functions of component i of ξ , $i = 1, \dots, m$. If we can find $r_i > 0, i = 1, \dots, m$ such that the following three conditions hold:

1. *The functions g_i are $(-r_i)$ -concave*
2. *The marginal distribution Functions F_i are generated by $(r_i + 1)$ -decreasing densities f_i with associated parameter t_i^**
3. *The copula C is log-exp concave on $[q, 1]$, where $q \in (0, 1)^m$ is defined as $q_i = F_i(t_i^*), i = 1, \dots, m$.*

Then the set $\{x \in \mathbb{R}^n : \mathbb{P}[\xi \leq g(x)] \geq p\}$ is convex for all $p > p^ := \max_{i=1, \dots, m} F(t_i^*)$.*

Upon exploiting the independence of the rows of a Gaussian Technology Matrix, one can derive a formulation that allows for an indirect application of the previous result. This, in turn, allows us to assert eventual convexity of feasible sets with Gaussian Technology Matrices as the following result shows:

Theorem 2.5.25 ([110]). *Let A be a $m \times n$ matrix having a non degenerate multivariate Gaussian density in \mathbb{R}^{nm} with independent rows. Let $b \geq 0$ be a fixed vector. The set $M(p) := \{x \in \mathbb{R}^n : \mathbb{P}[Ax \leq b] \geq p\}$ is convex for all $p > p^*$, where $p^* =$*

$\Phi(\max\{\sqrt{3}, u^*\})$. Here Φ is the one dimensional standard Gaussian distribution function and

$$u^* = \max_{i=1, \dots, m} 4\lambda_{max}^i [\lambda_{min}]^{-\frac{3}{2}} \|\mu_i\|.$$

Moreover, μ_i is the average vector of the i -th row and $\lambda_{min}^i, \lambda_{max}^i$ refer to the smallest and largest eigenvalues of its covariance matrix Σ_i .

Eventual convexity will receive further attention in Section 3.4.

2.6 (Local) Lipschitz Continuity

Consider the special case of (2.0.1) wherein $g(x, z) = h(x) - z$, for $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$. In this case the probability constraint (2.0.1) becomes

$$\mathbb{P}[h(x) \geq \xi] \geq p. \quad (2.6.1)$$

Constraints of the form (2.6.1) involve the distribution function of the random variable ξ . The following result provides conditions under which such a distribution function is (Locally) Lipschitz Continuous:

Theorem 2.6.1 ([51]). *Let $\xi \in \mathbb{R}^m$ be a random variable admitting a density (with respect to the Lebesgue Measure). The distribution function $F_\xi(x) := \mathbb{P}[\xi \leq x]$ is (locally) Lipschitz continuous if each of the marginal distributions of ξ is (locally) Lipschitz.*

The following result characterizes the situation in which the distribution function is globally Lipschitz:

Theorem 2.6.2 ([204, 100]). *Let $\xi \in \mathbb{R}^m$ be a random variable admitting a density (with respect to the Lebesgue Measure). The distribution function $F_\xi(x) := \mathbb{P}[\xi \leq x]$ is Lipschitz continuous if and only if each of the marginal distributions of ξ is essentially bounded.*

Upon combining the notion of generalized concavity and under the assumption that the random variable ξ is non-degenerate (i.e., has a non-random component) we can arrive at an ever stronger result.

Theorem 2.6.3 ([109]). *Let ξ be a random variable in \mathbb{R}^m with quasi-concave law. Its distribution function $F_\xi(x) := \mathbb{P}[\xi \leq x]$ is Lipschitz continuous if and only if none of its components $\xi_i, i = 1, \dots, m$ has zero variance.*

2.7 Differentiability of Probabilistic Constraints

An important question for numerical treatment of constraints of type (2.0.1) is differentiability. Indeed many optimization algorithms require a gradient. Differentiability statements of chance constraints of the type (2.0.1) come mainly in two forms. On one

hand we have very general differentiability statements providing rather complex characterizations. On the other hand we have practical characterizations for special cases. In the works dealing with such special cases one tries to reduce the computation of any component of a gradient to the computation of integrals for which efficient numerical sampling approaches are available. Such approaches will also receive a thorough discussion in Sections 3.2 and 3.3.

2.7.1 General Statements

Theorem 2.7.1 ([234, 235, 237], for similar work see also [146, 147]). *Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^k$ be a continuously differentiable function and let $\theta : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a continuously differentiable density. Pick moreover $1 \leq l < k$ arbitrarily. Assume moreover that*

1. *The set $\mu(x) := \{y \in \mathbb{R}^m : g(x, y) \leq 0\}$ is bounded in a neighbourhood U of some point \bar{x} .*
2. *At \bar{x} all constraints $g_i(\bar{x}, y) \leq 0$, $i = 1, \dots, k$ are active*
3. *One can find a continuous matrix function $H_l : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^{n \times m}$ satisfying*

$$H(x, y) \nabla_y g^l(x, y) + \nabla_x g^l(x, y) = 0,$$

where $g^l(x, y) = (g_1(x, y), \dots, g_l(x, y)) \in \mathbb{R}^l$.

4. *The matrix function H_l has a continuous partial derivative with respect to y .*
5. *The gradient $\nabla_y g_i(x, y) \neq 0$ on $\partial_i \mu(\bar{x}) := \mu(\bar{x}) \cap \{y \in \mathbb{R}^m : g_i(\bar{x}, y) = 0\}$.*
6. *For each $y \in \mu(\bar{x})$, the vectors $\nabla_y g_i(x, y)$, $i \in I(x, y) := \{j : g_j(x, y) = 0\}$ are linearly independent*

Then the mapping $\varphi(x) := \int_{\mu(x)} \theta(x, y) d\lambda(y) = \mathbb{P}[\mu(x)]$ is differentiable at \bar{x} and

$$\begin{aligned} \nabla_x \varphi(x) &= \int_{\mu(x)} \nabla_x \theta(x, y) + \operatorname{div}_y (\theta(x, y) H_l(x, y)) d\lambda(y) \\ &\quad - \sum_{i=l+1}^k \int_{\partial_i \mu(x)} \frac{\theta(x, y)}{\|\nabla_y g_i(x, y)\|} [\nabla_x g_i(x, y) + H_l(x, y) \nabla_y g_i(x, y)] dS, \end{aligned}$$

where λ is the Lebesgue measure on \mathbb{R}^m .

In Theorem 2.7.1 above one can select $1 \leq l < k$ in a way that is convenient for the application at hand. The special choices $l = 0$ and $l = k$ can also be made and lead to the following special cases (we refer to [236] for a modern version):

Theorem 2.7.2 ([193, 200, 232, 231, 233]). *With notation and conditions as in theorem 2.7.1. Let $l = 0$, then we have:*

$$\nabla_x \varphi(x) = \int_{\mu(x)} \nabla_x \theta(x, y) - \sum_{i=1}^k \int_{\partial_i \mu(x)} \frac{\theta(x, y)}{\|\nabla_y g_i(x, y)\|} \nabla_x g_i(x, y) dS.$$

If $l = k$ we have:

$$\nabla_x \varphi(x) = \int_{\mu(x)} \nabla_x \theta(x, y) + \operatorname{div}_y(\theta(x, y) H_k(x, y)) d\lambda(y)$$

2.7.2 Special Cases

In some special cases very efficient formulae for gradients can be established. These formulae can be derived upon considering specific densities for the random vector ξ of (2.0.1) and/or specifically structured chance constraints. When considering chance constraints with a separable structure as in (2.6.1), differentiability depends completely on that of the distribution function and the mapping h .

The following general result on differentiability of distribution functions can be derived. (For a comprehensive proof we refer to [100]).

Theorem 2.7.3 (e.g., [181]). *Let $\xi \in \mathbb{R}^m$ be a random vector with density $f_\xi : \mathbb{R}^m \rightarrow \mathbb{R}$. Fix any $\bar{z} \in \mathbb{R}^m$ and consider $F_\xi(z) := \mathbb{P}[\xi \leq z]$. If*

$$\varphi^{(i)}(t) := \int_{-\infty}^{\bar{z}_1} \dots \int_{-\infty}^{\bar{z}_{i-1}} \int_{-\infty}^{\bar{z}_{i+1}} \dots \int_{-\infty}^{\bar{z}_s} f_\xi(u_1, \dots, u_{i-1}, t, u_{i+1}, \dots, u_s) du_1 \dots du_{i-1} du_{i+1} \dots du_s,$$

is continuous for all $i = 1, \dots, s$, then $F_\xi(z)$ is partially differentiable at \bar{z} and

$$\frac{\partial F_\xi}{\partial z_i}(\bar{z}) = \varphi^{(i)}(\bar{z}_i). \quad (2.7.1)$$

Remark 2.7.4. *Stochastic Gradient Approaches for distribution functions of the above form can be found in [124, 125]. In energy applications, stochastic gradient approaches have been investigated for minimizing expected value objective functions (e.g., [59, 89]).*

An important specialization Theorem 2.7.3 can be obtained if we consider Gaussian random variables.

Lemma 2.7.5 ([175, 181]). *Let ξ be an m -dimensional Gaussian random vector with mean $\mu \in \mathbb{R}^m$ and positive definite variance-covariance matrix Σ . Then the distribution function $F_\xi(z) := \mathbb{P}[\xi \leq z]$ is continuously differentiable and in any fixed $z \in \mathbb{R}^m$ the following holds:*

$$\frac{\partial F_\xi}{\partial z_i}(z) = f_{\xi_i}(z_i) F_{\xi_{\setminus i}}(z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_m), i = 1, \dots, m. \quad (2.7.2)$$

Here $\tilde{\xi}(z_i)$ is a Gaussian random variable with mean $\hat{\mu} \in \mathbb{R}^{m-1}$ and $(m-1) \times (m-1)$ positive definite covariance matrix $\hat{\Sigma}$. Let D_m^i denote the m -th order identity matrix from which the i th row has been deleted. Then $\hat{\mu} = D_m^i(\mu + \Sigma_{ii}^{-1}(z_i - \mu_i)\Sigma_i)$ and $\hat{\Sigma} = D_m^i(\Sigma - \Sigma_{ii}^{-1}\Sigma_i\Sigma_i^\top)(D_m^i)^\top$, where Σ_i is the i -th column of Σ .

The above result requires the Gaussian random variable to be non-degenerate, i.e., have positive definite covariance matrix. This is an important restriction in many situations, in particular in network design when randomness on nodes is considered. In such applications the degeneracy occurs because we multiply a non-degenerate Gaussian random variable with a matrix having more lines than columns. The following result therefore allows for an important generalization of Lemma 2.7.5.

Theorem 2.7.6 ([106]). *Let A be a $k \times m$ matrix. Consider a linear inequality system $Ax \leq z$ and define*

$$\mathcal{I}(A, z) = \{I \subseteq \{1, \dots, k\} : \exists x \in \mathbb{R}^m \ a_i^\top x = z_i, i \in I, a_i^\top x < z_i, i \notin I\}.$$

Assume that $z \in \mathbb{R}^m$ is such that $Ax \leq z$ is nondegenerate (i.e., $\text{rank}\{a_i\}_{i \in I} = |I| \ \forall I \in \mathcal{I}(A, z)$). Let ξ be an m -dimensional Gaussian random vector with mean μ and positive definite variance-covariance matrix Σ . Then the mapping $\varphi(z) = \mathbb{P}[A\xi \leq z]$ is differentiable at z and

$$\frac{\partial \varphi}{\partial z_j}(z) = \begin{cases} 0 & \text{if } \{j\} \notin \mathcal{I}(A, z) \\ f_j(z_j)\mathbb{P}[A^{(j)}L^{(j)}\xi^{(j)} \leq z^{(j)} - A^{(j)}w^{(j)}] & \text{if } \{j\} \in \mathcal{I}(A, z) \end{cases} \quad (2.7.3)$$

Here $\xi^{(j)}$ is a centered $m-1$ dimensional Gaussian random variable with independent components, $A^{(j)}$ is obtained from A by deleting row j , $z^{(j)}$ is defined similarly. Moreover, $L^{(j)}$ is the Choleski matrix of $S^{(j)} := \Sigma - \frac{1}{a_j^\top \Sigma a_j} \Sigma a_j a_j^\top \Sigma$ (i.e., $S^{(j)} = L^{(j)}(L^{(j)})^\top$), $w^{(j)} = \mu + \frac{z_j - a_j^\top \mu}{a_j^\top \Sigma a_j} \Sigma a_j$ and f_j the one-dimensional Gaussian density with mean $\mu^\top a_j$ and variance $a_j^\top \Sigma a_j$. Finally the inequality system $A^{(j)}L^{(j)}y \leq z^{(j)} - A^{(j)}w^{(j)}$ is nondegenerate.

Other important special cases involve the computation of gradients of multi-variate Gamma [186] and Dirichlet Distributions [221, 181]

Theorem 2.7.7 ([186, 181]). *A multivariate Gamma distribution $\zeta \in \mathbb{R}^m$ is defined as $\zeta = A\eta$, where $\eta \in \mathbb{R}^{2^m-1}$ contains independent standard Gamma (with parameters ϑ_j) distributed components and A is a $m \times 2^m - 1$ matrix with non-zero columns, $A_{ij} \in \{0, 1\}$ for $i = 1, \dots, m, j = 1, \dots, 2^m - 1$. Define for each $i = 1, \dots, m$, $I_i \subseteq \{1, \dots, 2^m - 1\}$ as $I_i = \{j : A_{ij} = 1\}$. Then $\delta^i \in \mathbb{R}^{m-1}$,*

$$\delta_k^i = \frac{\sum_{j \in I_k \cap I_i} \eta_j}{\sum_{j \in I_i} \eta_j}, k = 1, \dots, i-1, i+1, \dots, m, \quad (2.7.4)$$

is an $m-1$ dimensional Dirichlet Distribution with parameters

$$\begin{aligned} \Theta_k &= \sum_{j \in I_k \cap I_i} \vartheta_j, k = 1, \dots, i-1, i+1, \dots, m \\ \Theta_{m+1} &= \sum_{j \in \cup_{k \neq i} I_k \setminus I_i} \vartheta_j, \end{aligned}$$

for each $i = 1, \dots, m$. Now $F(z) := \mathbb{P}[\zeta \leq z]$ is partially differentiable and

$$\frac{\partial F_\xi}{\partial z_i}(z) = \mathbb{P}[z_i \delta_k^i + \gamma_k \leq z_k \ \forall k \neq i] \frac{z_i^{\vartheta_i-1} e^{-z_i}}{\Gamma(\vartheta_i)}, \quad (2.7.5)$$

where $\gamma_k = \sum_{j \in I_k \cap \bar{I}_i} \eta_j$, $k = 1, \dots, i-1, i+1, \dots, m$, is an $m-1$ dimensional multivariate gamma distribution independent of δ^i .

Theorem 2.7.8 ([258, 181, 88]). Let $\xi \in \mathbb{R}^m$ have a multivariate Dirichlet distribution, i.e., have the density:

$$f(z_1, \dots, z_m) = \frac{\Gamma(\vartheta_1 + \dots + \vartheta_{m+1})}{\Gamma(\vartheta_1) \dots \Gamma(\vartheta_{m+1})} z_1^{\vartheta_1-1} \dots z_m^{\vartheta_m-1} (1 - \sum_{j=1}^m z_j)^{\vartheta_{m+1}-1},$$

on the unit simplex $z \in \Delta_m$ in dimension m (zero elsewhere). If

$$y^i = \left(\frac{z_1}{1-z_i}, \dots, \frac{z_{i-1}}{1-z_i}, \frac{z_{i+1}}{1-z_i}, \dots, \frac{z_m}{1-z_i} \right) \in \mathbb{R}^{m-1}$$

satisfies $y^{(1)} + y^{(2)} > 1$ or $y^{(1)} + y^{(2)} + y^{(3)} > 1$ (but $y^{(1)} + y^{(2)} \leq 1$) for the order-statistics $y^{(\cdot)}$, then $F(z) := \mathbb{P}[\xi \leq z]$ is partially differentiable at z and

$$\frac{\partial F_\xi}{\partial z_i}(z) = \mathbb{P}[\tilde{\xi}_k^i \leq y_k^i, \ \forall k \neq i] \frac{\Gamma(\vartheta_1 + \dots + \vartheta_{m+1})}{\Gamma(\vartheta_i) \Gamma(\sum_{j \neq i} \vartheta_j)} z_i^{\vartheta_i-1} (1 - z_i)^{\sum_{j \neq i} \vartheta_j-1}, \quad (2.7.6)$$

where $\tilde{\xi}^i$ has an $m-1$ dimensional Dirichlet distribution with parameters $\vartheta_1, \dots, \vartheta_{i-1}, \vartheta_{i+1}, \dots, \vartheta_{m+1}$.

2.8 Algorithms

In this section we provide a brief overview of known algorithms for solving optimization problems with chance constraints of type (2.0.1). Throughout this section we will assume that the resulting optimization problem is convex. For instance in a case where Theorem 2.5.15 can be applied. We refer to [182] for schemes for each of these algorithms. Algorithms will be discussed in details in sections 4.1 and 4.2.

For the purpose of discussion, let $c : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function, providing an alternative equivalent representation of constraint (2.0.1), i.e., $c(x) \leq 0$ if and only if (2.0.1) holds. For instance if g is (jointly)-quasi concave and ξ has a log-normal density, we may define $c(x) := \log(p) - \log(\mathbb{P}[g(x, \xi) \geq 0])$. The sub-gradient inequality then gives:

$$c(y) \geq c(x) + \langle g, y - x \rangle \ \forall y \in \mathbb{R}^n, \quad (2.8.1)$$

where $g \in \partial c(x)$ is an arbitrary sub-gradient of the convex function c at point $x \in \mathbb{R}^n$. When $y \in \mathbb{R}^n$ is feasible for (2.0.1) this implies that

$$0 \geq c(y) \geq c(x) + \langle g, y - x \rangle \ \forall y \text{ s.t. (2.0.1) holds.}$$

Now if $x \in \mathbb{R}^n$ is moreover chosen such that $c(x) = 0$, this reduces to

$$\langle g, y - x \rangle \leq 0, \quad (2.8.2)$$

which is a valid inequality for all feasible solutions $y \in \mathbb{R}^n$.

2.8.1 Feasible direction Methods

In this method the mapping c is assumed to be continuously differentiable. In a first stage a linear programming problem is solved in which constraint (2.8.2) appears with an additional auxiliary variable $z \in \mathbb{R}$ also appearing in a similarly formed equation involving the objective function:

$$\langle g, y - x \rangle - \Theta z \leq 0,$$

where Θ is an appropriately chosen constant and x our current iterate. Let (y^*, z^*) be the optimal solution of that linear program. In a second step we compute λ such that $x + \lambda(y^* - x)$ is feasible. The algorithm is stopped with optimal solution y^* whenever $z^* = 0$.

This approach is originally due to [266]. It should be noted however that Zoutendijk's method lacks the global convergence property as shown in [228]. We refer to the discussion in [153] for further information. A method of feasible directions was adapted to the chance constrained programming setting by [175], further tested in [48] and applied to a problem from the industry in [183].

2.8.2 SUMT or "penalty function methods"

Let $\varphi : \mathbb{R}^n \rightarrow [0, 1]$ be defined as $\varphi(x) = \mathbb{P}[g(x, \xi) \geq 0]$, where ξ and g are as in (2.0.1). In this approach we assume that φ is log-concave. It then follows that $x \mapsto \varphi(x) - p$ is also log-concave on $M(p)$, i.e., the feasible set for (2.0.1). The idea of the log-barrier approach is to add $-s^k \log(\varphi(x) - p)$ to the objective function. The sequence $\{s^k\}_{k \geq 0}$ of scalar multipliers is chosen strictly decreasing to zero. We then solve the (probabilistically) unconstrained optimization problem and update the multipliers iteratively. Details on SUMT approaches can be found in [73]. The application of this approach to chance constrained programming was suggested by [177]. An application and implementation can be found [194, 184].

2.8.3 Interior Point Methods

In this approach we add an additional slack variable w and the constraint $\varphi(x) - p = w$. We then form the classic logarithmic barrier function, wherein $-\mu \log(w)$ appears. The above slack constraints are dualized in the Lagrangian. We then apply the basic Interior Point approach: we establish the KKT conditions and use Newton's method to come up with search directions. This method has been investigated by many authors (e.g., [252], [205], [163], [259], etc...). One problem is that the Hessian often tends to become ill-conditioned. This might especially be true for probabilistic constraints. It is suggested in [182] to use the approaches of [157, 158] in order to counterbalance this effect.

2.8.4 Supporting Hyperplane Method

In this approach we require the explicit knowledge of a Slater point $x^s \in \mathbb{R}^n$. In particular we require $\varphi(x^s) > p$, where notation is as in section 2.8.2. The feasible set induced by constraint (2.0.1) needs to be convex as well. If x^k is the current iterate, we use the Slater point to form $x^c = \lambda x^k + (1 - \lambda)x^s$ in such a way that $c(x^c) = 0$, i.e., $\varphi(x^c) = p$. We then add (2.8.2) to the otherwise probabilistically unconstrained-optimization problem. This relaxation is then solved to produce the new iterate. More details on this approach are provided in sections 4.1 and section 5.1.

The idea of using cutting planes approaches for convex optimization dates back to the early 1960s ([123, 87, 36]). Cheney and Goldstein state in [36] that the key idea of their approach can be traced back to E. Remez (e.g., [195]). We can also refer to [253]. The cutting plane approach was adapted to the context of joint chance constrained programming by [185, 222].

2.9 Related Concepts

In this section we provide a brief overview of topics related to chance constrained programming but not directly used in this Thesis. They figure here for the sake of providing a global overview of the field.

2.9.1 Stability

A frequently formulated objection against using chance constraints of the type (2.0.1) concerns the availability of knowledge of ξ . It is then argued that either the ξ is unknown as a whole or at best its parameters are known up to some estimation error. It is then natural to investigate to what extent the solutions of optimization problems with constraints of type (2.0.1) are impacted by substituting a wrong law for ξ or picking wrong parameters. Stability results address this question.

As an application one can consider the reduction of the sample size when ξ has a discrete distribution and the sample would be too large for numerical treatment. Such a case is considered in [103, 104].

For a general treatment of stability in stochastic programming problems we refer to [201]. Specific stability results for optimization problems with chance constraints are derived in [62, 90, 107, 96, 98, 108, 118, 122, 203, 202, 212, 256].

In particular in the case of separable chance constraints of type (2.6.1) under a generalized concavity assumption on the distribution function, concavity of h and a compactness (non-emptiness) assumption on the optimal solution set of the unperturbed problem one can derive a Lipschitz continuity condition on the optimal values under small perturbations of ξ ([96][Theorem 1]). Under additional, somewhat restrictive conditions, one can also derive a Hausdorff-Hölder condition on the set of optimal solutions ([108]).

2.9.2 p -Efficient points

The concept of p -Efficient points generalizes the notion of quantile for 1-dimensional distribution functions. It is defined as follows:

Definition 2.9.1. *Let $\xi \in \mathbb{R}^m$ be a random vector with distribution function F_ξ . Let $p \in (0, 1)$ be given. A point $v \in \mathbb{R}^m$ is called a p -efficient point of the distribution function F_ξ if $F_\xi(v) \geq p$ and no $z \in \mathbb{R}^m$, $z \neq v$ can be found s.t. $z \leq v$ and $F_\xi(z) \geq p$.*

It is then clear that a chance constraint of the type (2.6.1) can be replaced with $h(x) \in \mathcal{E}_p$, where \mathcal{E}_p denotes the set of p -efficient points for the distribution function F_ξ . We refer to [51][pages 115-122] for some properties of the set \mathcal{E}_p .

The concept of p -efficient points was originally introduced for discrete distributions by [180]. An enumeration method and solution approach based on a cutting plane method was derived in [188]. The cone generation method wherein only a single p -efficient point is generated at each iteration was developed in [54] and further considered in [254]. A branch and bound algorithm was derived in [16] for the case wherein both x and ξ are integer valued. We also refer to [53] (and references therein) for a modern algorithm based on augmented Lagrangian methods using p -efficient points.

2.9.3 Safe Tractable Approximations

As highlighted throughout this section, theoretical and numerical treatment of constraints of type (2.0.1) might be involved. A very natural idea is then to replace the constraint (2.0.1) by constraints in a potentially higher dimensional space that imply feasibility for (2.0.1). More specifically, following [13], we define a "safe tractable approximation" as follows:

Definition 2.9.2. *Let $\mathcal{S} \subseteq \mathbb{R}^n \times \mathbb{R}^l$ be a convex set described by explicit convex functions. We call \mathcal{S} a safe tractable approximation of constraint (2.0.1) if for every $(x, v) \in \mathcal{S}$, x satisfies (2.0.1).*

Two important contributions in this field are [160, 15]. In the first work the so called "Bernstein" Approximation is derived for fairly general probability constraints under an independence assumption of the random vector ξ . In the second work the authors consider chance constraints on inequalities arising in Semi-Definite programming¹.

Frequently such an approximation is derived by using bounding techniques such as Hoeffding's bound ([115]). Such an approach is considered in [263, 264]. The obvious loss of optimality inherent to such an approach is a serious drawback. It can be attenuated by some ad-hoc parameter tweaking, such as deriving the safe tractable approximation under another security level p (in (2.0.1)). These approaches could be of interest to derive a first estimate of the optimal solution of an optimization problem with a constraint of type (2.0.1) cheaply. One would then consider a hybrid solution approach with a first

¹called linear matrix inequalities

stage wherein one derives a good estimate x_0 of the optimal solution. A second stage dealing with (2.0.1) in an appropriate way would then be applied to obtain the optimal solution.

2.9.4 Scenario Approximations / Sample Average Approximations

The key idea of these approaches is to replace (2.0.1) by a sampled version. To this end let ξ_1, \dots, ξ_N be an (iid) sample of the random vector ξ . We introduce auxiliary binary variables $z_i \in \{0, 1\}$ for $i = 1, \dots, N$ and look at the following approximation of (2.0.1):

$$\begin{aligned} g(x, \xi_i) &\geq (z_i - 1)Me & (2.9.1) \\ \frac{1}{N} \sum_{i=1}^N z_i &\geq q, \end{aligned}$$

where $q \geq p$, $M > 0$ is an appropriately chosen "big"-M constant and $e = (1, \dots, 1) \in \mathbb{R}^k$. Intuitively each z_i corresponds to an active constraint on sample value ξ_i when $z_i = 1$ and deactivated constraints whenever $z_i = 0$. Other than inherent difficulties due to "big"-M constraints this approach raises several questions. In particular to what extent can we expect a feasible solution of (2.9.1) to be feasible for (2.0.1). A second question naturally involves the required sample size N .

The Scenario Approximation refers to a situation wherein we pick $q = 1$, i.e., all binary variables in (2.9.1) are removed. This approach was developed by [28] and extended for large p by [159, 161]. Both works provide a result linking a given confidence level, q , and the sample size N . Very similar links are also exhibited in [42] in the context of sampling constraints in approximate dynamic programming. The scenario approximation approach can alternatively be interpreted as a version of Robust Optimization with a discrete uncertainty set. Heuristically discarding some elements of the sample has also been investigated in [30]. The Sample Average Approximation approach (i.e., picking $q < 1$ in (2.9.1)) was first investigated for Probabilistic constraints in [145]. The same approach was traditionally used for stochastic programs with expected value objective functions (see [215] and references therein, see also [168]). Hence [145] provide an extension of this concept to applications involving probabilistic constraints. In particular, a result linking a given confidence level, q , and the sample size N is given.

In a recent paper [144] an approach is presented for specially structured problems that allows one to do away with the big M -constraints in (2.9.1). Under additional assumptions on x , even the auxiliary binary variables need not be explicitly generated. This is investigated in [220]

Even though, in all these results, we can link the required sample size N , q and a confidence level for satisfying the actual constraint (2.0.1), N can be prohibitively large. We note that the number of constraints in (2.9.1) is equal to Nk . Now if $k = 100$ and $N = 10000$ this may be particularly troublesome for MIP / LP solvers, even if $g(x, z)$ is linear in x . Moreover, in a recent paper [102] shows that one can design problems

already in dimension 2, i.e., $n = 2$ in (2.0.1) wherein N has to be arbitrarily large in order to have a rough estimate of the optimal solution. The results are illustrated for the Gaussian distribution. This shows that such sample average approximations should be used with care.

Chapter 3

Theoretical Extensions

We begin this chapter by investigating to what extent the feasible set induced by a joint probabilistic constraint can be approximated by a generalization of individual probabilistic constraints. In principle an infinite set of such individual probabilistic constraints should be considered. This approach could be of interest if for a current set of constraints and current candidate solution we can readily identify a violated constraint, which can then be added to the problem. A second condition is that the feasible set is a better approximation of the feasible set induced by the joint probabilistic constraint than that generated by individual probabilistic constraints. An example shows that this second condition could be satisfied in specific applications. We will also derive (negative) results showing that this second point does not hold in general.

Sections 3.2 and 3.3 are devoted to deriving gradient Formulae for probabilistic constraints involving Gaussian random variables in one way or another. The provided formulae allow for a direct application of the same sampling approach used to evaluate the probability constraint we took the derivative of.

Whenever the feasible set for constraints of type (2.0.1) is convex when p is high enough, we will speak of eventual convexity. For separable constraints (2.0.4) and random variables allowing for a special "correlation" structure through the use of a Copula recent results [110, 111] on eventual convexity exist. The results in Section 3.4 provide an extension of the latter results on eventual convexity, by allowing for more Copulae and providing lower thresholds. Such lower thresholds allow us to assert convexity for probabilistically constrained feasible set with lower safety-levels.

3.1 Semi-Infinite Individual Chance Constraints

Consider the stochastic inequality system $g(x, \xi) \geq 0$ appearing in (2.0.1). It is then clear that for each fixed $\xi \in \mathbb{R}^m$ we have $g(x, \xi) \geq 0$ if and only if $v^\top g(x, \xi) \geq 0$ for all $v \in \mathbb{R}^k, v \geq 0$. This motivates the following definition:

Definition 3.1.1. *Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^k$ be a mapping and $\xi \in \mathbb{R}^m$ a random variable.*

Consider the probabilistic constraint

$$\mathbb{P}[g(x, \xi) \geq 0] \geq p, \quad (3.1.1)$$

and its feasible set $M(p) := \{x \in \mathbb{R}^n : \mathbb{P}[g(x, \xi) \geq 0] \geq p\}$. Let $v \in \mathbb{R}^k$, $v \geq 0$ be arbitrary. We call $x \in \mathbb{R}^n$ feasible for the v -Individual Chance Constraint (v -ICC) if and only if

$$\mathbb{P}[v^\top g(x, \xi) \geq 0] \geq p. \quad (3.1.2)$$

We say that $x \in \mathbb{R}^n$ is feasible for the semi-infinite individual chance constraint system if x is v -ICC feasible for each $v \in \mathbb{R}^k, v \geq 0$. We denote the latter feasible set by $M^{SI}(p) := \{x \in \mathbb{R}^n : (3.1.2) \text{ holds for all } v \in \mathbb{R}^k, v \geq 0\}$.

Definition 3.1.2. Under notation of Definition 3.1.1, we say that $x \in \mathbb{R}^n$ is individual chance constrained feasible (ICC-feasible) if and only if

$$\mathbb{P}[g_i(x, \xi) \geq 0] \geq p, i = 1, \dots, k. \quad (3.1.3)$$

We will denote this with $x \in M^I(p)$.

The following Lemma shows that Definition 3.1.1 might allow for a tighter approximation of $M(p)$ using v -Individual chance constraints.

Lemma 3.1.3. Consider the situation of Definition 3.1.1. We have :

$$M(p) \subseteq M^{SI}(p) \subseteq M^I(p). \quad (3.1.4)$$

Proof. The inclusion $M^{SI}(p) \subseteq M^I(p)$ follows trivially. Indeed when $x \in \mathbb{R}^n$ is ICC-feasible it is e_i -ICC feasible, where $e_i \in \mathbb{R}^k$ denotes the i -th standard unit vector and $i = 1, \dots, k$ is arbitrary. Now let $x \in \mathbb{R}^n$ be such that $x \in M(p)$. It then follows that a (measurable)-set $\Xi \subseteq \mathbb{R}^m$ can be found with $\mathbb{P}[\xi \in \Xi] \geq p$ and $g(x, z) \geq 0$ for all $z \in \Xi$. As a consequence, $v^\top g(x, z) \geq 0$ for all $z \in \Xi$ and $v \in \mathbb{R}^k, v \geq 0$. Therefore x is v -ICC feasible for each $v \in \mathbb{R}^k, v \geq 0$, i.e., $x \in M^{SI}(p)$. \square

The potential interest of Definition 3.1.1 is revealed by the following example:

Example 3.1.4. Let $\xi \in \mathbb{R}^2$ be a random variable with a discrete distribution taking values $(2, 1), (1, 1)$ and $(1, 2)$ with probability $\frac{1}{5}, \frac{3}{5}, \frac{1}{5}$ respectively. Consider the constraint $\mathbb{P}[\xi \leq x] \geq \frac{4}{5}$. It is then readily observed that $M^I(p) = (1, 1) + \mathbb{R}_+^2$ and $M(p) = (2, 1) + \mathbb{R}_+^2 \cup (1, 2) + \mathbb{R}_+^2$. Taking $v = (\frac{1}{2}, \frac{1}{2})$, it is clear that $\mathbb{P}[v^\top \xi \leq v^\top x] \geq \frac{4}{5}$ if and only if $v^\top x \geq \frac{3}{2}$. It readily follows that $M^{SI}(p) = (1, 1) + \mathbb{R}_+^2 \cap \{x \in \mathbb{R}^2 : v^\top x \geq \frac{3}{2}\} = \text{conv}(M(p))$. This example shows that in general the inclusions in Lemma 3.1.3 are strict.

The following Lemma is trivially derived upon noting that $x \in \mathbb{R}^n$ is v -ICC feasible for $v \in \mathbb{R}^k, v \geq 0$ if and only if x is λv -ICC feasible for an arbitrary $\lambda > 0$.

Lemma 3.1.5. Let $\|\cdot\|$ be an arbitrary norm in \mathbb{R}^k . The set $M^{SI}(p)$ of definition 3.1.1 is invariant when restricting ourselves in its definition to all $v \in \mathbb{R}^k, v \geq 0$ with $\|v\| = 1$.

The following results show that if the constraint of (3.1.1) is separable and ξ has a (non-degenerate) symmetric log-concave density, then $M^{SI}(p) = M^I(p)$.

Lemma 3.1.6. *Let A be a $m \times n$ deterministic matrix and let g appearing in (3.1.1) be defined as $g(x, z) = Ax - z$. Assume moreover that the random variable $\xi \in \mathbb{R}^m$ induces a non-degenerate probability measure \mathbb{P} , with symmetric strictly positive log-concave density. Then for all $p \in (\frac{1}{2}, 1)$ we have $M^{SI}(p) = M^I(p)$.*

Proof. Let $p \in (\frac{1}{2}, 1)$ be fixed but arbitrary. Under the assumptions on ξ it follows that there exists a so called floating body for \mathbb{P} ([51, Theorem 4.33],[151]). This means that there exists a convex compact set $C_p \subseteq \mathbb{R}^m$ such that

$$\mathbb{P}[v^\top \xi \geq s_{C_p}(v)] = 1 - p, \quad (3.1.5)$$

where $v \in \mathbb{R}^m$ is arbitrary and $s_{C_p}(v) = \sup_{h \in C_p} h^\top v$ is the support function of the set C_p . Since $\xi \in \mathbb{R}^m$ has a density (with respect to the Lebesgue measure) it does not allow for point masses. Hence we also have $\mathbb{P}[v^\top \xi \leq s_{C_p}(v)] = p$. We now claim that x is v -ICC feasible if and only if $v^\top(Ax) \geq s_{C_p}(v)$. To show this claim, fix $v \in \mathbb{R}^m, v \geq 0$ arbitrarily. If x is such that $v^\top(Ax) \geq s_{C_p}(v)$ then $\mathbb{P}[v^\top \xi \leq v^\top(Ax)] \geq \mathbb{P}[v^\top \xi \leq s_{C_p}(v)] = p$ and so x is v -ICC feasible. In contrast assume that x is v -ICC feasible, but that $v^\top(Ax) < s_{C_p}(v)$. We have made the assumption that the density of ξ is strictly positive. Hence the density of $v^\top \xi$ is also strictly positive. In particular on the interval $[v^\top(Ax), s_{C_p}(v)] \subseteq \mathbb{R}$. As a consequence of this and (3.1.5):

$$\mathbb{P}[v^\top \xi \geq v^\top(Ax)] > \mathbb{P}[v^\top \xi \geq s_{C_p}(v)] = 1 - p, \quad (3.1.6)$$

i.e., $\mathbb{P}[v^\top \xi \leq v^\top(Ax)] < p$. This contradicts that x is v -ICC feasible. We have thus shown our claim.

Let $v \in \mathbb{R}^m$ now be such that $\|v\|_1 = 1$ and $v \geq 0$. Let $x \in M^I(p)$ be arbitrary. From the above characterization it follows that $e_i^\top(Ax) \geq s_{C_p}(e_i)$ for $i = 1, \dots, m$, where e_i denotes the i -th standard unit vector of \mathbb{R}^m . Then,

$$v^\top Ax = \sum_{i=1}^m v_i(e_i^\top Ax) \geq \sum_{i=1}^m v_i(s_{C_p}(e_i)) \geq s_{C_p}\left(\sum_i v_i e_i\right),$$

where we have used the convexity of the mapping s_{C_p} . Using once again the above claim, we have shown that x is v -ICC feasible. Since v was arbitrary, upon using Lemma 3.1.5, we can conclude that $x \in M^{SI}(p)$. The result then follows upon using Lemma 3.1.3. \square

Lemma 3.1.7. *Let A be a $k \times m$ deterministic matrix. Assume moreover that the random variable $\xi \in \mathbb{R}^m$ induces a non-degenerate probability measure \mathbb{P} , with symmetric log-concave density. Then for each $p \in (\frac{1}{2}, 1)$, there exists a convex compact set $\hat{C}_p \subseteq \mathbb{R}^k$ such that*

$$\mathbb{P}[v^\top A\xi \geq s_{\hat{C}_p}(v)] = 1 - p, \quad (3.1.7)$$

for all $v \notin \ker A^\top$. Here $s_{\hat{C}_p} : \mathbb{R}^k \rightarrow \mathbb{R}$ is the support function of the set \hat{C}_p and $\ker A^\top$ denotes the null-space of the matrix A^\top .

Proof. Let $p \in (\frac{1}{2}, 1)$ be arbitrary. We can apply the floating body Theorem to the random variable ξ and thus establish equation (3.1.5) for a convex compact set $C_p \subseteq \mathbb{R}^m$. We now define $\hat{C}_p = AC_p \subseteq \mathbb{R}^k$. This set is clearly convex and compact. It remains to establish (3.1.7). We begin by noting that for any $v \in \mathbb{R}^k$,

$$s_{\hat{C}_p}(v) = \sup_{k \in \hat{C}_p} k^\top v = \sup_{h \in C_p} (Ah)^\top v = \sup_{h \in C_p} h^\top A^\top v. \quad (3.1.8)$$

It directly follows that $s_{\hat{C}_p}(v) = 0$ when $v \in \ker A^\top$. Similarly we get $v^\top A\xi = (A^\top v)^\top \xi = 0$. Fix an arbitrary $v \notin \ker A^\top$. From (3.1.8) and (3.1.5) we derive:

$$\mathbb{P}[v^\top (A\xi) \geq s_{\hat{C}_p}(v)] = \mathbb{P}[(A^\top v)^\top \xi \geq s_{C_p}(A^\top v)] = 1 - p,$$

since $A^\top v \in \mathbb{R}^m$ is arbitrary. □

We can now establish :

Theorem 3.1.8. *Let A and B be deterministic matrices of sizes $k \times m$ and $k \times n$ respectively. Let g appearing in (3.1.1) be defined as $g(x, z) = Bx - Az$. Assume moreover that the random variable $\xi \in \mathbb{R}^m$ induces a non-degenerate probability measure \mathbb{P} , with symmetric strictly positive log-concave density. Then for all $p \in (\frac{1}{2}, 1)$ we have $M^{SI}(p) = M^I(p)$.*

Proof. Under the assumptions of the Theorem, we can apply Lemma 3.1.7 to establish the existence of a compact convex set $\hat{C}_p \subseteq \mathbb{R}^k$ such that (3.1.7) holds. We now claim that $x \in \mathbb{R}^n$ is v -ICC feasible if and only if $v^\top (Bx) \geq s_{\hat{C}_p}(v)$ for all $v \in \mathbb{R}^k$. For any $v \in \ker A^\top$ this relation is directly established from the definition of v -ICC feasibility upon noting that $s_{\hat{C}_p}(v) = 0$ in that case. We can thus restrain ourselves to $v \notin \ker A^\top$. The proof of the claim is then identical to that of Lemma 3.1.6.

The claim having been established we can conclude the proof in a similar way as that of Lemma 3.1.6. □

Theorem 3.1.8 allows us to fully characterize the set $M^{SI}(p)$ for important constraints appearing in energy management applications:

Corollary 3.1.9. *Let g appearing in (3.1.1) be defined in such a way that (3.1.1) is equivalent with:*

$$\mathbb{P}[a + Ax \leq \xi \leq b + Bx] \geq p, \quad (3.1.9)$$

where $a, b \in \mathbb{R}^m$ and the $m \times n$ matrices A, B are deterministic. Let $\xi \in \mathbb{R}^m$ be a non-degenerate multivariate Gaussian random variable with mean μ and variance-covariance matrix Σ . Then for each $p > \frac{1}{2}$ we have $M^{SI}(p) = M^I(p)$.

Proof. The constraint appearing in (3.1.9) is equivalent with:

$$\mathbb{P}\left[\begin{pmatrix} I \\ -I \end{pmatrix} \tilde{\xi} \leq \begin{pmatrix} B & (b - \mu) \\ -A & (-a + \mu) \end{pmatrix} \tilde{x}\right] \geq p, \quad (3.1.10)$$

where $\tilde{x} \in \mathbb{R}^{n+1}$, $\tilde{x}_{n+1} = 1$ and $\tilde{\xi} \in \mathbb{R}^n$ is a centered multi-variate Gaussian random variable with positive definite covariance matrix Σ . The random vector $\tilde{\xi}$ and constraint (3.1.10) satisfy the conditions of Theorem 3.1.8. We thus establish that $\tilde{M}^{SI}(p) = \tilde{M}^I(p)$ as subsets of \mathbb{R}^{n+1} . In particular for each $\tilde{x} \in \tilde{M}^I(p)$ with $\tilde{x}_{n+1} = 1$, we have $\tilde{x} \in \tilde{M}^{SI}(p)$. We have thus shown that $M^{SI}(p) = M^I(p)$ as was claimed. \square

We end this section with an explicit characterization of bilateral individual chance constraints in the setting of (3.1.9) when $B = A$.

Lemma 3.1.10. *Let g appearing in (3.1.1) be defined in such a way that (3.1.1) is equivalent with:*

$$\mathbb{P}[a + Ax \leq \xi \leq b + Ax] \geq p, \quad (3.1.11)$$

where $a, b \in \mathbb{R}^m$ and the $m \times n$ matrix A are deterministic. Let $\xi \in \mathbb{R}^m$ be a non-degenerate multivariate Gaussian random variable with mean μ and variance-covariance matrix Σ . Then for each $p > \frac{1}{2}$ there exist $\underline{y}(p), \bar{y}(p) \in \mathbb{R}^m$ such that

$$M^I(p) = \{x \in \mathbb{R}^n : \underline{y}(p) \leq Ax \leq \bar{y}(p)\}. \quad (3.1.12)$$

Moreover the set $[\underline{y}(p), \bar{y}(p)]$ is bounded.

Proof. Upon redefining a and b we may assume without loss of generality that $\mu = 0$. A vector $x \in \mathbb{R}^n$ satisfies (3.1.11) if and only if there exists $y \in \mathbb{R}^m$, $y = Ax$ and $\mathbb{P}[a + y \leq \xi \leq b + y] \geq p$. It is thus sufficient to consider the case $A = I$. Let $i = 1, \dots, m$ be arbitrary but fixed and let $y \in \mathbb{R}^m$ be e_i -ICC feasible. By definition we have $\mathbb{P}[a_i + y_i \leq \xi_i \leq b_i + y_i] \geq p$. It then follows from $p > \frac{1}{2}$ that $a_i + y_i \leq 0$ and $b_i + y_i \geq 0$. Hence $y_i \in [-b_i, -a_i]$ and the latter set is bounded. Since ξ_i is a non-degenerate Gaussian random variable, it follows from Theorem 2.5.13 that the set of e_i -ICC feasible points is convex. As a matter of fact, even $M^I(p)$ is convex, since $i = 1, \dots, m$ is arbitrary. In particular if y_i^1 and y_i^2 (with $y_i^1 < y_i^2$) are two e_i -ICC feasible points it follows from convexity that any point in the interval $[y_i^1, y_i^2]$ is also e_i -ICC feasible. Since ξ_i does not have point-masses a smallest feasible y_i can be found, i.e., $\underline{y}(p)_i := \min \{y_i : \mathbb{P}[a_i + y_i \leq \xi_i \leq b_i + y_i] \geq p\}$ is well-defined. We can define $\bar{y}(p)_i$ similarly. The lemma has been shown. \square

The strength of Lemma 3.1.10 is that the box $[\underline{y}(p), \bar{y}(p)]$ can be computed numerically with very high precision with for instance a dichotomy procedure. For each $i = 1, \dots, m$, the above proof shows that $\underline{y}(p)_i \in [-b_i, -a_i]$. By iteratively dividing this interval in two and evaluating the Gaussian distribution function in dimension 1 we can compute $\underline{y}(p)_i$ up to (nearly) arbitrary precision. Lemma 3.1.10 has the advantage of providing an explicit characterization of $M^I(p)$ that does not require yet a weaker characterization involving individual unilateral constraints.

3.2 Efficient Gradient Formulae : bilateral separable probabilistic constraints

In energy management, probabilistic constraints of the type (2.0.1) arise naturally, but have a special structure. We refer to [243, 245] for an investigation of these structures. Often, the random inequality system $g(x, \xi) \geq 0$ appearing in (2.0.1) is a linear inequality system $A(\xi)x \leq b(\xi)$, where $A(\xi)$ and/or $b(\xi)$ are random. One even naturally encounters an inequality system of the form

$$a + Ax \leq \xi \leq b + Bx, \quad (3.2.1)$$

where $\xi \in \mathbb{R}^m$ is a random vector and the vectors $a, b \in \mathbb{R}^m$ and $m \times n$ matrices A, B are deterministic. A system of the form (3.2.1) is encountered in cascaded reservoir management. Then, ξ represents random inflows and the constraint the fact that the volume in a hydro-reservoir has to remain between a lower and upper bound. More details can be found in section 5.1.

The probabilistic constraint of the form (2.0.1) associated with system (3.2.1) is:

$$\mathbb{P}[a + Ax \leq \xi \leq b + Bx] \geq p. \quad (3.2.2)$$

A first question of interest is the differentiability with respect to x of the constraint appearing in (3.2.2). For instance if $\xi \in \mathbb{R}^m$ is assumed to follow a non-degenerate multivariate Gaussian distribution. Of course, such a bilateral system can be mapped to a unilateral one as follows:

$$\begin{pmatrix} I \\ -I \end{pmatrix} \xi \leq \begin{pmatrix} B \\ -A \end{pmatrix} x + \begin{pmatrix} b \\ -a \end{pmatrix}. \quad (3.2.3)$$

According to the reformulation (3.2.3), one might be tempted to apply Lemma 2.7.5 in order to derive a formula for the derivative. However the random vector

$$\eta := \begin{pmatrix} I \\ -I \end{pmatrix} \xi \in \mathbb{R}^{2m}$$

appearing in (3.2.3) is degenerate. Therefore, Lemma 2.7.5 can not be applied. Clearly the recent Theorem 2.7.6 would directly lead to a formula for the derivative. We report here an earlier derived result ([244]) that has the advantage of not requiring the evaluation of probabilities in dimension $2m$ contrary to the result appearing in Theorem 2.7.6.

Lemma 3.2.1. *Let ξ be an m -dimensional Gaussian random vector with mean $\mu \in \mathbb{R}^m$ and positive definite variance-covariance matrix Σ . We define the mapping $F_\xi(a, b) := \mathbb{P}[a \leq \xi \leq b]$ for any rectangle, i.e., $a \leq b$. Let D_m^i denote the m -th order identity matrix from which the i th row has been deleted. For each $y \in \mathbb{R}^m$, $1 \leq i \leq m$ and $z \in \mathbb{R}$ we define $y^{i(z)} = D_m^i(y + \Sigma_{i,i}^{-1}(z - y_i)\Sigma_i) \in \mathbb{R}^{m-1}$, where Σ_i is the i th column of Σ . Also define $\Sigma^{m(i)} = D_m^i(\Sigma - \Sigma_{i,i}^{-1}\Sigma_i\Sigma_i^\top)(D_m^i)^\top$. Moreover, let $f_{\nu,\sigma}(x)$ be the one-dimensional*

Gaussian density with mean ν and variance σ . Then for arbitrary but fixed $i = 1, \dots, m$, we have:

$$\begin{aligned}\frac{\partial}{\partial b_i} F_\xi(a, b) &= f_{\mu_i, \sigma_{ii}}(b_i) F_{\xi_i(b_i)}(D_m^i a, D_m^i b) \\ \frac{\partial}{\partial a_i} F_\xi(a, b) &= -f_{\mu_i, \sigma_{ii}}(a_i) F_{\xi_i(a_i)}(D_m^i a, D_m^i b),\end{aligned}$$

where $\xi_i(z) \sim \mathcal{N}(\mu^{c_i^i}(z), \Sigma^{c^m(i)}) \in \mathbb{R}^{m-1}$.

Proof. It is well known (e.g., [19]) that

$$F_\xi(a, b) = \sum_{j_1, \dots, j_m \in \{0,1\}} (-1)^{m + \sum_{k=1}^m j_k} \Phi_\xi(y_{j_1}, \dots, y_{j_m}), \quad (3.2.4)$$

where $\Phi_\xi : \mathbb{R}^m \rightarrow [0, 1]$ is the cumulative distribution function of ξ and

$$y_{j_k} = \begin{cases} a_k & \text{if } j_k = 0 \\ b_k & \text{if } j_k = 1. \end{cases}$$

Let $i = 1, \dots, m$ be arbitrary but fixed and consider the partial derivative of $F_\xi(a, b)$ with respect to b_i . Then according to (3.2.4) we have:

$$\frac{\partial}{\partial b_i} F_\xi(a, b) = \sum_{j_1, \dots, j_m \in \{0,1\}} (-1)^{m + \sum_{k=1}^m j_k} \frac{\partial}{\partial b_i} \Phi_\xi(y_{j_1}, \dots, y_{j_m}).$$

Now if $j_i = 0$ then $b_i \notin \{y_{j_1}, \dots, y_{j_m}\}$ and $\frac{\partial}{\partial b_i} \Phi_\xi(y_{j_1}, \dots, y_{j_m}) = 0$. On the other hand, if $j_i = 1$, we derive by applying Lemma 2.7.5 that:

$$\begin{aligned}\frac{\partial}{\partial b_i} \Phi_\xi(y_{j_1}, \dots, y_{j_m}) &= \frac{\partial}{\partial b_i} \Phi_\xi(y_{j_1}, \dots, y_{j_{i-1}}, b_i, y_{j_{i+1}}, \dots, y_{j_m}) \\ &= f_{\mu_i, \sigma_{ii}}(b_i) \Phi_{\xi_i(b_i)}(y_{j_1}, \dots, y_{j_{i-1}}, y_{j_{i+1}}, \dots, y_{j_m}),\end{aligned}$$

where $\xi_i(b_i) \sim \mathcal{N}(\mu^{c_i^i}(b_i), \Sigma^{c^m(i)}) \in \mathbb{R}^{m-1}$ is a Gaussian random variable. Upon combining this result with (3.2.4) we derive:

$$\begin{aligned}\frac{\partial}{\partial b_i} F_\xi(a, b) &= \\ &= \sum_{j_1, \dots, j_{i-1}, j_{i+1}, \dots, j_m \in \{0,1\}} (-1)^{m+1 + \sum_{k=1, k \neq i}^m j_k} f_{\mu_i, \sigma_{ii}}(b_i) \Phi_{\xi_i(b_i)}(y_{j_1}, \dots, y_{j_{i-1}}, y_{j_{i+1}}, \dots, y_{j_m}) \\ &= f_{\mu_i, \sigma_{ii}}(b_i) \sum_{j_1, \dots, j_{i-1}, j_{i+1}, \dots, j_m \in \{0,1\}} (-1)^{m-1 + \sum_{k=1, k \neq i}^m j_k} \Phi_{\xi_i(b_i)}(y_{j_1}, \dots, y_{j_{i-1}}, y_{j_{i+1}}, \dots, y_{j_m}) \\ &= f_{\mu_i, \sigma_{ii}}(b_i) F_{\xi_i(b_i)}(D_m^i a, D_m^i b).\end{aligned}$$

The asserted formula for the partial derivative with respect to a_i follows similarly upon noting that $a_i \notin \{y_{j_1}, \dots, y_{j_m}\}$ when $j_i = 1$. \square

This result can be readily extended to derive a Formula for the Hessian ([246]):

Lemma 3.2.2. *Let ξ be an m -dimensional Gaussian random vector with mean $\mu \in \mathbb{R}^m$ and positive definite variance-covariance matrix Σ . We define the mapping $F_\xi(a, b) := \mathbb{P}[a \leq \xi \leq b]$ for any rectangle, i.e., $a \leq b$. Let D_m^i denote the m -th order identity matrix from which the i th row has been deleted. For each $y \in \mathbb{R}^m$, $1 \leq i \leq m$ and $z \in \mathbb{R}$ we define $y^{c^{i,m}(z, \Sigma_i)} = D_m^i(y + \Sigma_{i,i}^{-1}(z - y_i)\Sigma_i) \in \mathbb{R}^{m-1}$, where Σ_i is the i th column of Σ . We will occasionally abbreviate this with $y^{c_1^i(z)}$. We also define*

$$y^{c_2^{i,j}(z,w)} = (y^{c^{i,m}(z, \Sigma_i)})^{c^{j,m-1}(w, \Sigma_j^{c^m(i)})},$$

where we have defined $\Sigma^{c^m(i)} = D_m^i(\Sigma - \Sigma_{i,i}^{-1}\Sigma_i\Sigma_i^\top)(D_m^i)^\top$. We define $\xi^{c_1^i(z)}$ as the Gaussian random variable with mean $\mu^{c_1^i(z)}$ and covariance matrix $\Sigma^{c^m(i)}$. In a similar way, we define $\xi^{c_2^{i,j}(z,w)}$ as the Gaussian random variable with mean $\mu^{c_2^{i,j}(z,w)}$ and covariance matrix $\Sigma^{c_2^{i,j}} := D_{m-1}^j(\Sigma^{c^m(i)} - (\Sigma_{j,j}^{c^m(i)})^{-1}\Sigma_j^{c^m(i)}(\Sigma_j^{c^m(i)})^\top)(D_{m-1}^j)^\top$, where $\Sigma_j^{c^m(i)}$ denotes the j -th column of $\Sigma^{c^m(i)}$. The following holds, for $j = \hat{j}$ if $\hat{j} < i$ and $j = \hat{j} - 1$ if $\hat{j} > i$:

$$\begin{aligned} \frac{\partial^2}{\partial a_j \partial a_i} F_\xi(a, b) &= f_{\mu_j^{c_1^i(a_i)}, \Sigma_{j,j}^{c^m(i)}}(a_j) f_{\mu_i, \Sigma_{i,i}}(a_i) F_{\xi^{c_2^{i,j}(a_i, a_j)}}(D_{m-1}^j D_m^i a, D_{m-1}^j D_m^i b) \quad \forall \hat{j} \neq i \\ \frac{\partial^2}{\partial b_j \partial a_i} F_\xi(a, b) &= \begin{cases} -f_{\mu_j^{c_1^i(a_i)}, \Sigma_{j,j}^{c^m(i)}}(b_j) f_{\mu_i, \Sigma_{i,i}}(a_i) F_{\xi^{c_2^{i,j}(a_i, b_j)}}(D_{m-1}^j D_m^i a, D_{m-1}^j D_m^i b) & \forall \hat{j} \neq i \\ 0 & \hat{j} = i \end{cases} \\ \frac{\partial^2}{\partial b_j \partial b_i} F_\xi(a, b) &= f_{\mu_j^{c_1^i(b_i)}, \Sigma_{j,j}^{c^m(i)}}(b_j) f_{\mu_i, \Sigma_{i,i}}(b_i) F_{\xi^{c_2^{i,j}(b_i, b_j)}}(D_{m-1}^j D_m^i a, D_{m-1}^j D_m^i b) \quad \forall \hat{j} \neq i, \end{aligned}$$

where $f_{\nu, \sigma}(x)$ is the one-dimensional Gaussian density with mean ν and variance σ . Moreover, whenever $j = i$ and z is a or b we have:

$$\begin{aligned} \frac{\partial}{\partial z_i}(b_i - a_i) \frac{\partial^2}{\partial z_i^2} F_\xi(a, b) &= -\frac{z_i - \mu_i}{\Sigma_{i,i}} f_{\mu_i, \Sigma_{i,i}}(z_i) F_{\xi^{c_1^i(z_i)}}(D_m^i a, D_m^i b) \\ &\quad - f_{\mu_i, \Sigma_{i,i}}(z_i) (D_m^i \Sigma_{i,i}^{-1} \Sigma_i)^\top (\nabla_{\tilde{a}} F_{\xi^{c_1^i(z_i)}}(\tilde{a}, \tilde{b}) + \nabla_{\tilde{b}} F_{\xi^{c_1^i(z_i)}}(\tilde{a}, \tilde{b})), \end{aligned}$$

where $\tilde{a} = D_m^i a - \mu^{c_1^i(z_i)}$, $\tilde{\xi}^{c_1^i(z_i)} = \xi^{c_1^i(z_i)} - \mu^{c_1^i(z_i)}$ and \tilde{b} is defined similarly.

Proof. The formula for the cross derivatives follow from a straight-forward second application of Lemma 3.2.1. The diagonal terms are more subtle to derive and require the following reformulation:

$$\begin{aligned} F_{\xi^{c_1^i(z_i)}}(D_m^i a, D_m^i b) &= \mathbb{P}\left(D_m^i a \leq \xi^{c_1^i(z_i)} \leq D_m^i b\right) \\ &= \mathbb{P}\left(D_m^i a - \mu^{c_1^i(z_i)} \leq \xi^{c_1^i(z_i)} - \mu^{c_1^i(z_i)} \leq D_m^i b - \mu^{c_1^i(z_i)}\right) \\ &= F_{\tilde{\xi}^{c_1^i(z_i)}}(\tilde{a}, \tilde{b}). \end{aligned}$$

In particular one obtains for $\tilde{a}(z_i) = \tilde{a}$

$$\begin{aligned} \tilde{a} &= D_m^i a - \mu^{c_1^i(z_i)} = D_m^i a - D_m^i(\mu + \Sigma_{i,i}^{-1}(z_i - \mu_i)\Sigma_i) \\ &= D_m^i(a - \mu + \Sigma_{i,i}^{-1}\mu_i\Sigma_i) - D_m^i \Sigma_{i,i}^{-1} z_i \Sigma_i, \end{aligned}$$

which together with the following identity

$$\frac{\partial}{\partial z_i} F_{\xi^{c_1^i}(z_i)}(\tilde{a}(z_i), \tilde{b}(z_i)) = \nabla_{\tilde{a}} F_{\xi^{c_1^i}(z_i)}(\tilde{a}, \tilde{b}) D_{z_i} \tilde{a}(z_i) + \nabla_{\tilde{b}} F_{\xi^{c_1^i}(z_i)}(\tilde{a}, \tilde{b}) D_{z_i} \tilde{b}(z_i),$$

an application of the chain-rule and the already established formula for 1st derivatives gives the proposition. \square

Considering the probability constraint (3.2.2), its derivative can be computed according to the following Corollary:

Corollary 3.2.3. *Let $\varphi : \mathbb{R}^n \rightarrow [0, 1]$ be defined as $\varphi(x) := \mathbb{P}[a + Ax \leq \xi \leq b + Bx]$, where $\xi \in \mathbb{R}^m$ is a Gaussian random variable with mean $\mu \in \mathbb{R}^m$ and positive definite variance-covariance matrix Σ . Moreover a, b, A, B are as in (3.2.2). Then the mapping φ is twice differentiable and we have:*

$$\begin{aligned} \nabla \varphi &= \nabla_a F_\xi(a, b)^\top A + \nabla_b F_\xi(a, b)^\top B \\ \nabla^2 \varphi &= A^\top \nabla_{aa}^2 F_\xi(a, b) A + A^\top \nabla_{ab}^2 F_\xi(a, b) B + B^\top \nabla_{ba}^2 F_\xi(a, b) A + B^\top \nabla_{bb}^2 F_\xi(a, b) B, \end{aligned}$$

where F_ξ is defined as in Lemma 3.2.1.

One can also compute the derivative of a chance constraint wherein the inequality system $g(x, \xi) \geq 0$ of (2.0.1) is a linear inequality system $A(\xi)x \leq b$ with $A(\xi)$ having a multivariate Gaussian distribution. The following results was derived in [247] in a (slightly) more general form

Theorem 3.2.4. *Consider the mapping $\varphi : \mathbb{R}^n \rightarrow [0, 1]$ defined as $\varphi(x) = \mathbb{P}[A(\xi)x \leq \alpha(x)]$, where $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a differentiable mapping and $A(\xi)$ a $m \times n$ matrix, with components following a multi-variate Gaussian random variable in $\mathbb{R}^{m \times n}$ with mean (matrix) $\mu \in \mathbb{R}^{m \times n}$ and positive definite $mn \times mn$ covariance matrix Σ . Let $x \neq 0$ be arbitrary and fixed. Then $\eta(x) := A(\xi)x \in \mathbb{R}^m$ is a non-degenerate multivariate Gaussian random variable with mean $\mu(x) \in \mathbb{R}^m$, covariance matrix $\Sigma(x)$ and correlation matrix $R(x)$.*

Define $\tilde{D}(x) \in \mathbb{R}^m$ as the diagonal of $\Sigma(x)$ and $D(x)$ as the diagonal matrix with vector $\tilde{D}(x)^{-\frac{1}{2}}$ on its diagonal. The latter operation is to be understood element by element. Define the mapping $\beta : \mathbb{R}^n \rightarrow \mathbb{R}^m$ as $\beta(x) = D(x)(\alpha(x) - \mu(x))$. Then φ is differentiable at x and we have :

$$\begin{aligned} \nabla \varphi(x) &= - \sum_{i=1}^m f_{0, R_{ii}(x)}(\beta_i(x)) \mathbb{P}[\tilde{\xi}^i(x) \leq D_m^i \beta(x)] (\alpha_i(x) - \mu_i(x)) \frac{1}{2} \Sigma_{ii}^{-\frac{3}{2}}(x) \Sigma^{ii} x \\ &+ \sum_{i=1}^m f_{0, R_{ii}(x)}(\beta_i(x)) \mathbb{P}[\tilde{\xi}^i(x) \leq D_m^i \beta(x)] \Sigma_{ii}^{-\frac{1}{2}}(x) (\nabla \alpha_i(x) - \mu_i) \\ &+ \sum_{1 \leq i < j \leq m} \Sigma_{ii}^{-\frac{3}{2}}(x) \Sigma_{jj}^{-\frac{3}{2}}(x) [\Sigma_{ii}(x) \Sigma_{jj}(x) \Sigma^{ii} x - \frac{1}{2} \Sigma_{ij}(x) [\Sigma_{jj}(x) \Sigma^{ii} x + \Sigma_{ii}(x) \Sigma^{jj} x] \cdot \\ &\cdot f_{0, \tilde{R}_{j-1, j-1}^i(x)}(\beta_j(x) - \tilde{\mu}_{j-1}^i(x)) f_{0, R_{ii}(x)}(\beta_i(x)) \mathbb{P}[\tilde{\xi}^{ij}(x) \leq D_{m-1}^{j-1} D_m^i (\beta(x) - \tilde{\mu}^i(x))] \end{aligned}$$

where for $i = 1, \dots, m$, D_m^i is defined as in Lemma 2.7.5, μ_i denotes the i -th row of matrix μ and $\tilde{\xi}^i(x) \in \mathbb{R}^{m-1}$ is a multi-variate Gaussian random variable with mean $\tilde{\mu}^i(x) := D_m^i R_{ii}^{-1}(x) \beta_i(x) R_i(x)$ and covariance matrix

$$\tilde{R}^i(x) := D_m^i (R(x) - R_{ii}^{-1}(x) R_i(x) R_i(x)^\top) (D_m^i)^\top,$$

where $R_i(x)$ denotes the i -th column of the matrix $R(x)$. Moreover $f_{0,\sigma}(z)$ denotes the density of a 1-dimensional Gaussian random variable with variance σ evaluated in $z \in \mathbb{R}$. Finally, for $i, j = 1, \dots, m$ $\tilde{\xi}^{ij}(x) \in \mathbb{R}^{m-2}$ is a non-degenerate Gaussian random variable with mean $D_{m-1}^{j-1} ((\tilde{R}_{j-1,j-1}^i(x))^{-1} (\beta_j(x) - \tilde{\mu}_{j-1}^i(x)) \tilde{R}_{j-1}^i(x)$ and covariance matrix

$$D_{m-1}^{j-1} (\tilde{R}^i(x) - \tilde{R}_{j-1,j-1}^i(x))^{-1} \tilde{R}_{j-1}^i(x) (\tilde{R}_{j-1}^i(x))^\top.$$

Again $\tilde{R}_{j-1}^i(x)$ denotes the $j-1$ -st column of matrix $\tilde{R}^i(x)$. The matrix Σ^{ij} appearing in the above expression is an $n \times n$ matrix defined as $\Sigma_{lk}^{ij} = \Sigma_{(i-1)n+l, (j-1)n+k}$, $1 \leq l, k \leq n$.

Proof. It is convenient to reformulate φ . To this end we introduce the mapping $T : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$:

$$T(x) = \begin{pmatrix} x^\top & 0 & \cdots & 0 \\ 0 & x^\top & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & x^\top \end{pmatrix}.$$

We also define the following matrix operation $\cdot^\odot : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{mn}$, defined as:

$$A^\odot = \begin{pmatrix} A_{11} \\ \vdots \\ A_{1n} \\ A_{21} \\ \vdots \\ A_{mn} \end{pmatrix}.$$

It then readily follows that $\varphi(x) = \mathbb{P}[T(x)\xi^\odot \leq \alpha(x)]$. Now $\eta(x) := A(\xi)x = T(x)\xi^\odot$ follows a multi-variate Gaussian distribution with mean $\mu(x) = T(x)\mu^\odot = \mu x \in \mathbb{R}^m$ and $m \times m$ covariance matrix $\Sigma(x) = T(x)\Sigma T(x)^\top$. This means that for any $1 \leq i, j \leq m$ one has $\Sigma_{ij}(x) = x^\top \Sigma^{ij} x$, where Σ^{ij} is the $n \times n$ matrix defined as $\Sigma_{lk}^{ij} = \Sigma_{(i-1)n+l, (j-1)n+k}$, $1 \leq l, k \leq n$.

Let $R(x) = D(x)\Sigma(x)D(x)$. It is then easily observed that $R(x)$ is a correlation matrix and that φ is equivalent with $\varphi(x) = \Phi^{R(x)}(\beta(x))$, where $\Phi^R(z)$ denotes the m dimensional multivariate Gaussian distribution function with correlation matrix R evaluated at $z \in \mathbb{R}^m$. Let $w \in \mathbb{R}^m$ be such that $w \neq 0$ but otherwise arbitrary. It is easily observed that $T(x)^\top w = (w_1 x, \dots, w_m x) \in \mathbb{R}^{mn}$ and as a consequence for $x \neq 0$, $\Sigma(x)$ is positive definite. We have thus shown that $\eta(x)$ is non-degenerate as claimed.

Now define the mapping $\gamma : \mathbb{R}^{m \times m} \times \mathbb{R}^m \rightarrow [0, 1]$ as $\gamma(R, z) = \Phi^R(z)$. This mapping is continuously differentiable (when R is positive definite) and we establish in this manner

$$\nabla \varphi(x) = \sum_{i=1}^m \frac{\partial \gamma}{\partial z_i}(R(x), \beta(x)) \nabla \beta_i(x) + \sum_{1 \leq i < j \leq m} \frac{\partial \gamma}{\partial R_{ij}}(R(x), \beta(x)) \nabla R_{ij}(x). \quad (3.2.5)$$

Let $i = 1, \dots, m$ be arbitrary. Differentiating $\beta_i(x)$ leads to

$$\nabla \beta_i(x) = (\alpha_i(x) - \mu_i(x)) \nabla \Sigma_{ii}^{-\frac{1}{2}}(x) + \Sigma_{ii}^{-\frac{1}{2}}(x) (\nabla \alpha_i(x) - \nabla \mu_i(x)). \quad (3.2.6)$$

We also readily derive

$$\begin{aligned} \nabla \mu_i(x) &= (\mu_{i1}, \dots, \mu_{in}) = \mu_i \\ \nabla \Sigma_{ii}^{-\frac{1}{2}}(x) &= -\frac{1}{2} \Sigma_{ii}^{-\frac{3}{2}}(x) \nabla \Sigma_{ii}(x) = -\frac{1}{2} \Sigma_{ii}^{-\frac{3}{2}}(x) \Sigma^{ii} x. \end{aligned}$$

The derivative $\frac{\partial \gamma}{\partial z_i}(R(x), \beta(x))$ is easily computed by applying Lemma 2.7.5 and we establish that

$$\frac{\partial \gamma}{\partial z_i}(R(x), \beta(x)) = f_{0, R_{ii}(x)}(\beta_i(x)) \mathbb{P}[\tilde{\xi}^i(x) \leq D_m^i \beta(x)], \quad (3.2.7)$$

where $\tilde{\xi}^i(x) \in \mathbb{R}^{m-1}$ is a multi-variate Gaussian random variable with mean $\tilde{\mu}^i(x)$ and covariance matrix $\tilde{R}^i(x)$.

One can note that $R_{ij}(x) = \Sigma_{ii}^{-\frac{1}{2}}(x) \Sigma_{jj}^{-\frac{1}{2}}(x) \Sigma_{ij}(x)$ for $i, j \in \{1, \dots, m\}$. Taking the derivative of this expression leads to

$$\nabla R_{ij}(x) = \Sigma_{ii}^{-\frac{3}{2}}(x) \Sigma_{jj}^{-\frac{3}{2}}(x) [\Sigma_{ii}(x) \Sigma_{jj}(x) \Sigma^{ii} x - \frac{1}{2} \Sigma_{ij}(x) [\Sigma_{jj}(x) \Sigma^{ii} x + \Sigma_{ii}(x) \Sigma^{jj} x]]$$

Let $1 \leq i < j \leq m$ be given. From Gupta's formula [94] we derive that

$$\frac{\partial \gamma}{\partial R_{ij}}(R(x), \beta(x)) = \frac{\partial^2 \gamma}{\partial z_i \partial z_j}(R(x), \beta(x)). \quad (3.2.8)$$

The latter second derivative can be computed by using Lemma 3.2.2 above by setting $a = -\infty$. We then establish that

$$\begin{aligned} \frac{\partial^2 \gamma}{\partial z_i \partial z_j}(R(x), \beta(x)) &= \\ &= f_{0, \tilde{R}_{j-1, j-1}^i(x)}(\beta_j(x) - \tilde{\mu}_{j-1}^i(x)) f_{0, R_{ii}(x)}(\beta_i(x)) \mathbb{P}[\tilde{\xi}^{ij}(x) \leq D_{m-1}^{j-1} D_m^i (\beta(x) - \tilde{\mu}^i(x))]. \end{aligned}$$

Combining the above inequalities then yields the result. \square

3.3 Efficient Gradient Formulae : nonlinear probabilistic constraints with Gaussian and Gaussian-like distributions

A probabilistic constraint is an inequality of the type

$$\mathbb{P}[g(x, \xi) \geq 0] \geq p, \quad (3.3.1)$$

where g is a mapping defining a (random) inequality system and ξ is an s -dimensional random vector defined on some probability space $(\Omega, \mathcal{A}, \mathbb{P})$. The constraint (3.3.1) expresses the requirement that a decision vector x is feasible if and only if the random inequality system $g(x, \xi) \geq 0$ is satisfied at least with probability $p \in [0, 1]$.

From a formal viewpoint, (3.3.1) is a conventional inequality constraint $\alpha(x) \geq p$ with $\alpha(x) := \mathbb{P}[g(x, \xi) \geq 0]$. On the other hand, a major difficulty arises from the fact that typically no analytical expression is available for α . All one can hope for, in general, are tools for numerically approximating α . Beyond crude Monte Carlo estimation of the probability defining α , there exist a lot of more efficient approaches based, for instance, on graph-theoretical arguments [25], variance reduction [224], Quasi-Monte-Carlo (QMC) techniques or sparse grid numerical integration [82]. It seems, however, that such approaches are most successful when exploiting the special model structure (i.e., the mapping g and the distribution of ξ). For instance, in the special case of separable constraints $g(x, \xi) = x - \xi$ and of ξ having a regular Gaussian distribution (such that α reduces to a multivariate Gaussian distribution function), Genz [83, 84] developed a numerical integration scheme combining separation and reordering of variables with randomized QMC. Using this method, one may compute values of the Gaussian distribution function at fairly good precision in reasonable time even for a few hundred random variables. A similar technique has been proposed for the multivariate Student (or T-) distribution [84]. The numerical evaluation of other multivariate distribution functions such as Gamma or exponential distribution has been discussed, e.g., in [223, 166].

For an efficient solution of probabilistically constrained problems via numerical nonlinear optimization it is evidently not sufficient to calculate just functional values of α , one also has to have access to gradients of α . The need to calculate gradients of probability functions has been recognized a long time ago and has given rise to many papers on representing such gradients (e.g., [146], [234], [126], [170], [80]). In the separable case with Gaussian distribution mentioned above, it is well-known [181, p. 203], that partial derivatives of α can be reduced analytically to function values $\tilde{\alpha}$ of a Gaussian distribution with different parameters. This fact has three important consequences: first it allows one to employ the same efficient method (e.g. by Genz) available for values of Gaussian distribution functions in order to compute gradients simultaneously; second, doing so, the error in calculating $\nabla \alpha$ can be controlled by that in calculating α [101]; third, the mentioned analytic relation can be applied inductively, in order to get similar analytic relations between function values and higher-order derivatives. Fortunately, this very special circumstance can be extended to more general models: it has been demonstrated in [244, 247, 106] how for general linear probabilistic constraints $\alpha(x) := \mathbb{P}[T(x)\xi \leq$

$a(x)] \geq p$ under Gaussian distribution and with possibly nonregular, nonquadratic matrix $T(x)$ not only the computation of α (which is evident) but also of ∇a can be analytically reduced to the computation of Gaussian distribution functions. Combining appropriately these ideas with Genz' code and an SQP solver, it is possible to solve corresponding optimization problems for Gaussian random vectors in dimension of up to a few hundred (where the dimension of the decision vector x is less influential). Applications to various problems of power management can be found, e.g., in [244, 247, 246, 5, 106]. It seems that the same approach can be elaborated also for the multivariate Student distribution, whereas it would work for the Log-normal distribution only in the special case of $\alpha(x) = \mathbb{P}[b(x) \leq \xi \leq a(x)]$.

When considering models which are nonlinear in ξ , a reduction to distribution functions seems not to be possible any more. In that case, another approach, the so-called spherical-radial decomposition of Gaussian random vectors (see, e.g., [84]) seems to be promising both for calculating function values and gradients of α . More precisely, let ξ be an m -dimensional random vector normally distributed according to $\xi \sim \mathcal{N}(0, R)$ for some correlation matrix R . Then, $\xi = \eta L \zeta$, where $R = LL^T$ is the Cholesky decomposition of R , η has a chi-distribution with m degrees of freedom and ζ has a uniform distribution over the Euclidean unit sphere

$$\mathbb{S}^{m-1} := \left\{ z \in \mathbb{R}^m \mid \sum_{i=1}^m z_i^2 = 1 \right\}$$

of \mathbb{R}^m . As a consequence, for any Lebesgue measurable set $M \subseteq \mathbb{R}^m$ its probability may be represented as

$$\mathbb{P}[\xi \in M] = \int_{v \in \mathbb{S}^{m-1}} \mu_\eta(\{r \geq 0 : rLv \cap M \neq \emptyset\}) d\mu_\zeta, \quad (3.3.2)$$

where μ_η and μ_ζ are the laws of η and ζ , respectively. This probability can be numerically computed by employing an efficient sampling scheme on \mathbb{S}^{m-1} proposed by Deák [49, 50]. More generally, one may approximate the integral

$$\int_{v \in \mathbb{S}^{m-1}} h(v) d\mu_\zeta \quad (3.3.3)$$

for any Lebesgue measurable function $h : \mathbb{S}^{m-1} \rightarrow \mathbb{R}$. In particular, for

$$h(v) := \mu_\eta(\{r \geq 0 : rLv \cap M \neq \emptyset\}),$$

we obtain the probability (3.3.2). In this section, we will show how - with different functions $h(v)$ - the same efficient sampling scheme can be employed in order to simultaneously compute derivatives of this probability with respect to an exterior parameter. The results may serve as a basis for a numerical treatment of nonlinear convex probabilistic constraints with Gaussian and alternative distributions via nonlinear optimization.

The formula, for the Gaussian case, that we provide in this paper was provided in a somewhat similar form in [58, Section 9.2], but without proof. A formula similar in idea

is given in [206] under an implicit assumption that $M(x) := \{z \in \mathbb{R}^m : g(x, z) \leq 0\}$, appearing in (3.3.2) is bounded, $0 \in \text{int } M(x)$ and an assumption that the gradients of g with respect to z are "non-degenerate" on the boundary of $M(x)$. That latter means that these gradients are not orthogonal to any ray starting at 0 hitting the boundary of $M(x)$. In the proof, differentiation and integration are interchanged without further comments. In the later work [207] the same authors provide a formula identical to ours. Again for the Gaussian case only. They moreover explicitly assume that $M(x)$ is bounded and that $0 \in \text{int } M(x)$. They also request that the gradients of g with respect to z are non-degenerate in the above way and make an additional μ_ζ zero measure assumption. The last assumption means that the set of directions $v \in \mathbb{S}^{m-1}$ hitting the boundary of $M(x)$ in points where more than one component of g is active, has zero measure. This last assumption is needed to derive a multi-variate version of the here provided formula. However, the boundedness assumption is restrictive. For example, it rules out the use of the formula for distribution functions. In this paper most work is devoted to the case when $M(x)$ is not assumed to be bounded. We moreover show that the gradients of g with respect to z are "non-degenerate" in the above way. If we would make the same zero-measure assumption as in [207] our results would moreover directly extend to the multi-variate case. However such a zero-measure assumption can not hold in general as can be observed in [109, Figure 1]. We will therefore restrain ourselves, for now, to the case wherein g has a single component only.

In Section 3.3.1, a rigorous justification for differentiating under the integral sign will be given. Doing so, we arrive at sufficient conditions for continuous differentiability of probability functions in the concave and Gaussian case as well as at an explicit integrand in (3.3.3). In Section 3.3.2, the obtained results are applied to various examples involving Gaussian and alternative distributions. Particular attention is paid to the multivariate Student distribution.

3.3.1 A gradient formula for parameter-dependent Gaussian probabilities in the convex case

In the following, we assume that $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ is a continuously differentiable function which is concave with respect to the second argument. We define

$$\varphi(x) := \mathbb{P}[g(x, \xi) \geq 0], \quad (3.3.4)$$

where $\xi \sim \mathcal{N}(0, R)$.

Remark 3.3.1. *We recall that convex sets are Lebesgue measurable so that the probabilities in (3.3.4) are well-defined by virtue of ξ having a density.*

Remark 3.3.2. *If ξ has a general nondegenerate Gaussian distribution, i.e., $\xi \sim \mathcal{N}(\mu, \Sigma)$ for some mean vector $\mu \in \mathbb{R}^m$ and some positive definite covariance matrix Σ of order (m, m) , then one may define $\tilde{\xi} := D(\xi - \mu)$, where D is the diagonal matrix with elements $\Sigma_{ii}^{-1/2}$. Then, clearly, $\tilde{\xi} \sim \mathcal{N}(0, R)$, where R is the correlation matrix associated*

with Σ . Defining $\tilde{g} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ as

$$\tilde{g}(x, z) := g(x, D^{-1}z + \mu),$$

(3.3.4) can be rewritten as

$$\varphi(x) = \mathbb{P} \left[\tilde{g}(x, \tilde{\xi}) \geq 0 \right],$$

where \tilde{g} has the same properties as g (it is continuously differentiable and concave with respect to the second argument). Therefore, in (3.3.4), we may indeed assume without loss of generality, that $\xi \sim \mathcal{N}(0, R)$.

By (3.3.2) and (3.3.4), we have, for all $x \in \mathbb{R}^n$, that

$$\varphi(x) = \int_{v \in \mathbb{S}^{m-1}} \mu_\eta(\{r \geq 0 : g(x, rLv) \geq 0\}) d\mu_\zeta = \int_{v \in \mathbb{S}^{m-1}} e(x, v) d\mu_\zeta \quad (3.3.5)$$

for

$$e(x, v) := \mu_\eta(\{r \geq 0 : g(x, rLv) \geq 0\}) \quad \forall x \in \mathbb{R}^n \quad \forall v \in \mathbb{S}^{m-1}. \quad (3.3.6)$$

According to the possibility of evaluating (3.3.3) for instance by Deak's method, we can obtain a value $\varphi(x)$ for each fixed x . We now address the computation of $\nabla\varphi$. It is convenient to introduce the following two mappings $F, I : \mathbb{R}^n \rightrightarrows \mathbb{S}^{m-1}$ of directions with finite and infinite intersection length:

$$\begin{aligned} F(x) &:= \{v \in \mathbb{S}^{m-1} | \exists r > 0 : g(x, rLv) = 0\} \\ I(x) &:= \{v \in \mathbb{S}^{m-1} | \forall r > 0 : g(x, rLv) \neq 0\}. \end{aligned}$$

The following Lemma collects some elementary properties needed later:

Lemma 3.3.3. *Let $x \in \mathbb{R}^n$ be such that $g(x, 0) > 0$. Then,*

1. $v \in I(x)$ if and only if $g(x, rLv) > 0$ for all $r > 0$.
2. $F(x) \cup I(x) = \mathbb{S}^{m-1}$.
3. For $v \in F(x)$ let $r > 0$ be such that $g(x, rLv) = 0$. Then,

$$\langle \nabla_z g(x, rLv), Lv \rangle \leq -\frac{g(x, 0)}{r}.$$

4. If $v \in I(x)$ then $e(x, v) = 1$, where e is defined in (3.3.6).

Proof. 1. follows from the continuity of g and 2. is evident from the definitions. The convexity of $-g$ with respect to the second argument yields

$$\begin{aligned} \frac{1}{2}r \langle \nabla_z g(x, rLv), Lv \rangle &= \left\langle -\nabla_z g(x, rLv), \frac{1}{2}rLv - rLv \right\rangle \leq g(x, rLv) - g\left(x, \frac{1}{2}rLv\right) \\ &= -g\left(x, \frac{1}{2}rLv\right) \leq -\frac{1}{2}g(x, 0) - \frac{1}{2}g(x, rLv) = -\frac{1}{2}g(x, 0). \end{aligned}$$

This proves 3. If $v \in I(x)$ then $e(x, v) = \mu_\eta(\mathbb{R}_+) = 1$ because \mathbb{R}_+ is the support of the chi-distribution. Therefore, 4. holds true. \square

Next, we provide a local representation of the factor r as a function of x and v :

Lemma 3.3.4. *Let (x, v) be such that $g(x, 0) > 0$ and $v \in F(x)$. Then, there exist neighbourhoods U of x and V of v as well as a continuously differentiable function $\rho^{x,v} : U \times V \rightarrow \mathbb{R}_+$ with the following properties:*

1. *For all $(x', v', r') \in U \times V \times \mathbb{R}_+$ the equivalence $g(x', r'Lv') = 0 \Leftrightarrow r' = \rho^{x,v}(x', v')$ holds true.*
2. *For all $(x', v') \in U \times V$ one has the gradient formula*

$$\nabla_x \rho^{x,v}(x', v') = -\frac{1}{\langle \nabla_z g(x', \rho^{x,v}(x', v')Lv'), Lv' \rangle} \nabla_x g(x', \rho^{x,v}(x', v')Lv').$$

Proof. By definition of $F(x)$ we have that $g(x, rLv) = 0$ for some $r > 0$. According to 3. in Lemma 3.3.3, we have that

$$\langle \nabla_z g(x, rLv), Lv \rangle \leq -\frac{g(x, 0)}{r} < 0.$$

This allows to apply the Implicit Function Theorem to the equation $g(x, rLv) = 0$ and to derive the existence of neighbourhoods U of x , V of v and W of r along with a continuously differentiable function $\rho^{x,v} : U \times V \rightarrow W$, such that the equivalence

$$g(x', r'Lv') = 0, (x', v', r') \in U \times V \times W \Leftrightarrow r' = \rho^{x,v}(x', v'), (x', v') \in U \times V \quad (3.3.7)$$

holds true. By continuity of $\rho^{x,v}$, we may shrink the neighbourhoods U and V such that $\rho^{x,v}$ maps into \mathbb{R}_+ and we may further shrink U such that $g(x', 0) > 0$ for all $x' \in U$. Now, assume that $g(x', r^*Lv') = 0$ holds true for some $(x', v', r^*) \in U \times V \times (\mathbb{R}_+ \setminus W)$. Then, by ' \Leftrightarrow ' in (3.3.7), $g(x', \rho^{x,v}(x', v')Lv') = 0$, where $\rho^{x,v}(x', v') \in W$. Consequently, $r^* \neq \rho^{x,v}(x', v')$. On the other hand, $r^*, \rho^{x,v}(x', v') \in \mathbb{R}_+$. This contradicts the concavity of g with respect to the second argument and the fact that $g(x', 0) > 0$. It follows that in (3.3.7) W may be replaced by \mathbb{R}_+ which proves 1. In particular, we have that $g(x', \rho^{x,v}(x', v')Lv') = 0$ for all $(x', v') \in U \times V$, which after differentiation gives the formula in 2. \square

The preceding Lemma allows us to observe the following:

Lemma 3.3.5. *Let $x \in \mathbb{R}^n$ be such that $g(x, 0) > 0$. Then,*

1. *If $v \in F(x)$ then there exist neighbourhoods U of x and V of v such that $e(x', v') = F_\eta(\rho^{x,v}(x', v'))$ for all $(x', v') \in U \times V$, where e is defined in (3.3.6), F_η is the cumulative distribution function of the chi-distribution with m degrees of freedom and $\rho^{x,v}$ refers to the resolving function introduced in Lemma 3.3.4.*
2. *If $v \in I(x)$ then $\rho^{x_k, v_k}(x_k, v_k) \rightarrow \infty$ for any sequence $(x_k, v_k) \rightarrow (x, v)$ with $v_k \in F(x_k)$.*

Proof. By 1. in Lemma 3.3.4, we have for all (x', v') that $g(x', \rho^{x',v}(x', v')Lv') = 0$ and $g(x', r'Lv') \neq 0$ for all $r' \in \mathbb{R}_+$ with $r' \neq \rho^{x',v}(x', v')$. Now, (3.3.6) implies that

$$e(x', v') = \mu_\eta([0, \rho^{x',v}(x', v')]) = F_\eta(\rho^{x',v}(x', v')) - F_\eta(0) \quad \forall (x', v') \in U \times V.$$

Now, 1. follows upon observing that the chi-density is zero for negative arguments, whence $F_\eta(0) = 0$. Next, let $v \in I(x)$ and $(x_k, v_k) \rightarrow (x, v)$ with $v_k \in F(x_k)$. If not $\rho^{x_k, v_k}(x_k, v_k) \rightarrow \infty$, then there exists a converging subsequence $\rho^{x_{k_l}, v_{k_l}}(x_{k_l}, v_{k_l}) \rightarrow r$ for some $r \geq 0$. Since $g(x, 0) > 0$, we have that $g(x_{k_l}, 0) > 0$ for l sufficiently large. This allows us to apply Lemma 3.3.4 to the points (x_{k_l}, v_{k_l}) , and so we infer from 1. in this Lemma that $g(x_{k_l}, \rho^{x_{k_l}, v_{k_l}}(x_{k_l}, v_{k_l})Lv_{k_l}) = 0$ for all l sufficiently large. By continuity of g we derive the contradiction $g(x, rLv) = 0$ with our assumption $v \in I(x)$. This proves 2. \square

Corollary 3.3.6. *The function $e : \mathbb{R}^n \times \mathbb{S}^{m-1} \rightarrow \mathbb{R}$ defined in (3.3.6) is continuous at any $(x, v) \in \mathbb{R}^n \times \mathbb{S}^{m-1}$ such that $g(x, 0) > 0$.*

Proof. Let $(x, v) \in \mathbb{R}^n \times \mathbb{S}^{m-1}$ with $g(x, 0) > 0$ be arbitrarily given. Referring to the sets $F(x)$ and $I(x)$ characterized in Lemma 3.3.3, there are two possibilities: if $v \in F(x)$, then the function $\rho^{x,v}$ is defined on a neighbourhood of (x, v) and is continuous there by Lemma 3.3.4. Moreover, in this case, e has the representation given in 1. of Lemma 3.3.5. But with the cumulative distribution function F_η of the chi-distribution being continuous, e is continuous too at (x, v) as a composition of continuous mappings. If, in contrast, $v \notin F(x)$, then $v \in I(x)$ by 2. of Lemma 3.3.3. From 4. of the same Lemma we know that $e(x, v) = 1$. Consider an arbitrary sequence $(x_k, v_k) \rightarrow (x, v)$ with $v_k \in \mathbb{S}^{m-1}$. Since $g(x, 0) > 0$, we have that $g(x_k, 0) > 0$ for k sufficiently large. Assume that not $e(x_k, v_k) \rightarrow 1$. Then, there is a subsequence (x_{k_l}, v_{k_l}) and some $\varepsilon > 0$ such that for all l

$$|e(x_{k_l}, v_{k_l}) - 1| \geq \varepsilon. \quad (3.3.8)$$

By 4. in Lemma 3.3.3, $v_{k_l} \notin I(x_{k_l})$, whence $v_{k_l} \in F(x_{k_l})$ for all l due to $v_{k_l} \in \mathbb{S}^{m-1}$ and 2. in Lemma 3.3.3. Then, $\rho^{x_{k_l}, v_{k_l}}(x_{k_l}, v_{k_l}) \rightarrow \infty$ by 2. of Lemma 3.3.5. Since F_η is the distribution function of a random variable, it satisfies the relation $\lim_{t \rightarrow \infty} F_\eta(t) = 1$. Consequently, we may invoke 1. of Lemma 3.3.5 in order to verify that

$$\lim_{l \rightarrow \infty} e(x_{k_l}, v_{k_l}) = \lim_{l \rightarrow \infty} F_\eta(\rho^{x_{k_l}, v_{k_l}}(x_{k_l}, v_{k_l})) = 1.$$

This contradicts (3.3.8) and, hence, again by 4. in Lemma 3.3.3,

$$\lim_{k \rightarrow \infty} e(x_k, v_k) = 1 = e(x, v).$$

This proves continuity of e at (x, v) . \square

Corollary 3.3.7. *For any $x \in \mathbb{R}^n$ with $g(x, 0) > 0$ and $v \in F(x)$ the partial derivative w.r.t x of the function $e : \mathbb{R}^n \times \mathbb{S}^{m-1} \rightarrow \mathbb{R}$ defined in (3.3.6) exists and is given by*

$$\nabla_x e(x, v) = -\frac{\chi(\rho^{x,v}(x, v))}{\langle \nabla_z g(x, \rho^{x,v}(x, v))Lv, Lv \rangle} \nabla_x g(x, \rho^{x,v}(x, v)Lv)$$

where χ is the density of the chi-distribution with m degrees of freedom and $\rho^{x,v}$ refers to the function introduced in Lemma 3.3.4.

Proof. By 1. in Lemma 3.3.5 we have that $e(x', v') = F_\eta(\rho^{x,v}(x', v'))$ for all x' in a neighbourhood of x and v' in a neighbourhood of v . Differentiation with respect to x yields

$$\nabla_x e(x', v') = \chi(\rho^{x,v}(x', v')) \nabla_x \rho^{x,v}(x', v') \quad (3.3.9)$$

due to $F'_\eta(\tau) = \chi(\tau)$ for $\tau > 0$. In particular, $\nabla_x e(x, v) = \chi(\rho^{x,v}(x, v)) \nabla_x \rho^{x,v}(x, v)$. Now, the assertion follows from 2. in Lemma 3.3.4. \square

Next, we prove a relation which is the key to some desired continuity properties.

Definition 3.3.8. Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a differentiable function. We say that g satisfies the **polynomial growth condition** at x if there exist constants $C, \varkappa > 0$ and a neighbourhood $U(x)$ such that

$$\|\nabla_x g(x', z)\| \leq \|z\|^\varkappa \quad \forall x' \in U(x) \quad \forall z : \|z\| \geq C.$$

Lemma 3.3.9. Let x be such that $g(x, 0) > 0$ and that g satisfies the polynomial growth condition at x . Consider any sequence $(x_k, v_k) \rightarrow (x, v)$ for some $v \in I(x)$ such that $v_k \in F(x_k)$. Then,

$$\lim_{k \rightarrow \infty} \nabla_x e(x_k, v_k) = 0.$$

Proof. First observe that $\rho^{x_k, v_k}(x_k, v_k) \rightarrow \infty$ by 2. in Lemma 3.3.5. Referring to the neighbourhood $U(x)$ from Definition 3.3.8, we verify that for k sufficiently large

$$\|\nabla_x g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k)\| \leq [\rho^{x_k, v_k}(x_k, v_k)]^\varkappa \|Lv_k\|^\varkappa \leq \|L\|^\varkappa [\rho^{x_k, v_k}(x_k, v_k)]^\varkappa \quad (3.3.10)$$

(recall that $\|v_k\| = 1$ due to $v_k \in F(x_k)$). Moreover, by continuity of g , there exists some $\delta_1 > 0$ such that $g(x_k, 0) \geq \delta_1 > 0$ for k sufficiently large. Since $g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k) = 0$ (see 1. in Lemma 3.3.4), 3. in Lemma 3.3.3 provides that

$$\langle \nabla_z g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k), Lv_k \rangle \leq -\frac{g(x_k, 0)}{\rho^{x_k, v_k}(x_k, v_k)}.$$

Therefore,

$$\langle \nabla_z g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k), Lv_k \rangle \leq -\delta_1 [\rho^{x_k, v_k}(x_k, v_k)]^{-1} < 0. \quad (3.3.11)$$

And as a consequence

$$|\langle \nabla_z g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k), Lv_k \rangle| \geq \delta_1 [\rho^{x_k, v_k}(x_k, v_k)]^{-1} > 0. \quad (3.3.12)$$

Using the definition $\chi(y) = \delta_2 y^{m-1} e^{-y^2/2}$ of the density of the chi-distribution with m degrees of freedom (where $\delta_2 > 0$ is an appropriate factor), we may combine Corollary 3.3.7 with (3.3.10) and (3.3.12) in order to derive that

$$\begin{aligned} \|\nabla_x e(x_k, v_k)\| &= \left\| \frac{\chi(\rho^{x_k, v_k}(x_k, v_k))}{\langle \nabla_z g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k), Lv_k \rangle} \nabla_x g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k) \right\| \leq \\ &\delta_1^{-1} \rho^{x_k, v_k}(x_k, v_k) \cdot \delta_2 [\rho^{x_k, v_k}(x_k, v_k)]^{m-1} e^{-[\rho^{x_k, v_k}(x_k, v_k)]^2/2} \cdot \|L\|^\varkappa [\rho^{x_k, v_k}(x_k, v_k)]^\varkappa \quad (3.3.13) \\ &\delta_1^{-1} \delta_2 \|L\|^\varkappa [\rho^{x_k, v_k}(x_k, v_k)]^{\varkappa+m} e^{-[\rho^{x_k, v_k}(x_k, v_k)]^2/2} \rightarrow_k 0, \end{aligned}$$

where the last limit follows from $\rho^{x_k, v_k}(x_k, v_k) \rightarrow \infty$ and the fact that $y^\alpha e^{-y^2/2} \rightarrow 0$ for $y \rightarrow \infty$, where $\alpha > 0$ is an arbitrary constant. This proves our assertion. \square

Remark 3.3.10. One may observe from the proof of Lemma 3.3.9 that a weaker growth condition than that in Definition 3.3.8 (involving an exponential term) would suffice for proving the same result. One could for instance use the following exponential growth condition:

Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a differentiable function. We say that g satisfies the **exponential growth condition** at x if there exist constants $\delta_0, C > 0$ and a neighbourhood $U(x)$ such that

$$\|\nabla_x g(x', z)\| \leq \delta_0 \exp(\|z\|) \quad \forall x' \in U(x) \quad \forall z : \|z\| \geq C.$$

and observe that the key estimate (3.3.13) of Lemma 3.3.9 becomes

$$\|\nabla_x e(x_k, v_k)\| \leq \delta_0 \delta_1^{-1} \delta_2 [\rho^{x_k, v_k}(x_k, v_k)]^m e^{-[\rho^{x_k, v_k}(x_k, v_k)]^2 / 2} e^{\|L\| \rho^{x_k, v_k}(x_k, v_k)}.$$

The same conclusion then easily follows.

We do not put the emphasis on the weakest possible form of the growth condition but rather on its simplicity. It should be noted however that each of the following results requiring the polynomial growth condition hold upon requiring the above exponential growth condition instead.

Corollary 3.3.11. Let x be such that $g(x, 0) > 0$ and that g satisfies the polynomial growth condition at x . Then, for any $v \in \mathbb{S}^{m-1}$ the partial derivative w.r.t x of the function e exists at (x, v) and is given by

$$\nabla_x e(x, v) = \begin{cases} -\frac{\chi(\rho^{x, v}(x, v))}{\langle \nabla_z g(x, \rho^{x, v}(x, v))Lv, Lv \rangle} \nabla_x g(x, \rho^{x, v}(x, v))Lv & \text{if } v \in F(x) \\ 0 & \text{else} \end{cases}$$

where χ is the density of the chi-distribution with m degrees of freedom and $\rho^{x, v}$ refers to the function introduced in Lemma 3.3.4.

Proof. Thanks to Corollary 3.3.7 and to 2. in Lemma 3.3.3 it is sufficient to show that $\nabla_x e(x, v) = 0$ for $v \in I(x)$. We shall show that, for any $i \in \{1, \dots, m\}$

$$\lim_{t \uparrow 0} \frac{e(x + tu_i, v) - e(x, v)}{t} = 0, \quad (3.3.14)$$

where u_i is the i -th canonical unit vector in \mathbb{R}^n . In exactly the same way one can show that the corresponding limit for $t \downarrow 0$ equals zero. Altogether, this will prove that $\nabla_x e(x, v) = 0$. Assume that (3.3.14) is wrong. Since $e(x, v) = 1$ (by 4. in Lemma 3.3.3) and $e(x + tu_i, v) \leq 1$ for all t (by definition of e as a probability in (3.3.6)), it follows that the quotient in (3.3.14) is always non-positive and, thus, negation of (3.3.14) implies the existence of some $\varepsilon > 0$ and of a sequence $t_k \uparrow 0$ such that

$$\frac{e(x + t_k u_i, v) - e(x, v)}{t_k} \geq \varepsilon. \quad (3.3.15)$$

In particular, $v \in F(x + t_k u_i)$ for all k because otherwise $v \in I(x + t_k u_i)$ and so $e(x + t_k u_i, v) = 1$ (again by 4. in Lemma 3.3.3), thus contradicting (3.3.15). We may also

assume that $g(x + t_k u_i, 0) > 0$ for all k . Now, fix an arbitrary k and define (recall that $t_k < 0$)

$$\alpha := \inf \{ \tau \in [t_k, 0] \mid e(x + \tau u_i, v) = 1 \}.$$

Due to $e(x, v) = 1$ we have that $\alpha \leq 0$. On the other hand, $e(x + t_k u_i, v) < 1$ and the continuity of e (see Corollary 3.3.6) provide that $\alpha > t_k$. We infer that $e(x + \tau u_i, v) < 1$ for all $\tau \in [t_k, \alpha)$ and, hence,

$$v \in F(x + \tau u_i) \quad \forall \tau \in [t_k, \alpha) \quad (3.3.16)$$

(once more by 2. and 4. in Lemma 3.3.3). But then, the function

$$\beta(\tau) := e(x + \tau u_i, v)$$

is differentiable for all $\tau \in (t_k, \alpha)$ by virtue of Corollary 3.3.7 and its derivative is given by

$$\beta'(\tau) = \langle \nabla_x e(x + \tau u_i, v), u_i \rangle.$$

Therefore, the mean value theorem guarantees the existence of some $\tau_k^* \in (t_k, \alpha)$ such that

$$\beta'(\tau_k^*) = \frac{\beta(\alpha) - \beta(t_k)}{\alpha - t_k}$$

or equivalently

$$\langle \nabla_x e(x + \tau_k^* u_i, v), u_i \rangle = \frac{e(x + \alpha u_i, v) - e(x + t_k u_i, v)}{\alpha - t_k}.$$

By continuity of e and by definition of α , we have that $e(x + \alpha u_i, v) = 1 = e(x, v)$, whence, by $t_k < \alpha \leq 0$,

$$\langle \nabla_x e(x + \tau_k^* u_i, v), u_i \rangle = \frac{e(x, v) - e(x + t_k u_i, v)}{\alpha - t_k} \geq \frac{e(x, v) - e(x + t_k u_i, v)}{-t_k} \geq \varepsilon,$$

where the last relation follows from (3.3.15). Now, since k was arbitrarily fixed, we have constructed a sequence τ_k^* such that $t_k < \tau_k^* \leq 0$ such that

$$\langle \nabla_x e(x + \tau_k^* u_i, v), u_i \rangle \geq \varepsilon \quad \forall k. \quad (3.3.17)$$

Since $t_k \uparrow 0$, we also have that $\tau_k^* \uparrow 0$. Moreover, $v \in F(x + \tau_k^* u_i)$ by (3.3.16). Due to our assumption that g satisfies the polynomial growth condition at x and due to $v \in I(x)$, Lemma 3.3.9 yields that $\lim_{k \rightarrow \infty} \nabla_x e(x_k, v) = 0$ which contradicts (3.3.17). This proves our Corollary. \square

Corollary 3.3.12. *Let x be such that $g(x, 0) > 0$ and that g satisfies the polynomial growth condition at x . Then, for any $v \in \mathbb{S}^{m-1}$ the partial derivative $\nabla_x e$ is continuous at (x, v) .*

Proof. Let $x \in \mathbb{R}^n$ with $g(x, 0) > 0$ and $v \in \mathbb{S}^{m-1}$ be arbitrarily given. Let also $(x_k, v_k) \rightarrow (x, v)$ be an arbitrary sequence with $v_k \in \mathbb{S}^{m-1}$. If $v \in F(x)$, then relation (3.3.9) holds true locally around (x, v) . In particular, for k large enough,

$$\nabla_x e(x_k, v_k) = \chi(\rho^{x,v}(x_k, v_k)) \nabla_x \rho^{x,v}(x_k, v_k) \rightarrow \chi(\rho^{x,v}(x, v)) \nabla_x \rho^{x,v}(x, v) = \nabla_x e(x, v),$$

where the convergence follows from the continuity of the chi-density and of $\nabla_x \rho^{x,v}$ as a result of Lemma 3.3.4. Hence, in case of $v \in F(x)$, $\nabla_x e$ is continuous at (x, v) . Now, assume in contrast that $v \in I(x)$. Then, $\nabla_x e(x, v) = 0$ by Corollary 3.3.11. Now, assume that $\nabla_x e(x_k, v_k)$ does not converge to zero. Then, $\|\nabla_x e(x_{k_l}, v_{k_l})\| \geq \varepsilon$ for some subsequence and some $\varepsilon > 0$. Then, $v_{k_l} \in F(x_{k_l})$ for all l because otherwise $v_{k_l} \in I(x_{k_l})$ and, thus, $\nabla_x e(x_{k_l}, v_{k_l}) = 0$ due to Corollary 3.3.11 (applied to x_{k_l} rather than x ; observe that the condition $g(x, 0) > 0$ and the polynomial growth condition at x are open conditions, hence continue to hold true for the x_{k_l}). Now, Lemma 3.3.9 yields the contradiction

$$\lim_{l \rightarrow \infty} \nabla_x e(x_{k_l}, v_{k_l}) = 0$$

with $\|\nabla_x e(x_{k_l}, v_{k_l})\| \geq \varepsilon$. This proves our Corollary. \square

Now we are in a position to state our main result:

Theorem 3.3.13. *Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a continuously differentiable function which is concave with respect to the second argument. Consider the probability function φ defined in (3.3.4), where $\xi \sim \mathcal{N}(0, R)$ has a standard Gaussian distribution with correlation matrix R . Let the following assumptions be satisfied at some \bar{x} :*

1. $g(\bar{x}, 0) > 0$.
2. g satisfies the polynomial growth condition at \bar{x} (Def. 3.3.8).

Then, φ is continuously differentiable on a neighbourhood U of \bar{x} and it holds that

$$\nabla \varphi(x) = - \int_{v \in F(x)} \frac{\chi(\rho^{x,v}(x, v))}{\langle \nabla_z g(x, \rho^{x,v}(x, v)) Lv, Lv \rangle} \nabla_x g(x, \rho^{x,v}(x, v)) Lv \, d\mu_\zeta(v) \quad \forall x \in U. \quad (3.3.18)$$

Here, μ_ζ is the law of the uniform distribution over \mathbb{S}^{m-1} , χ is the density of the chi-distribution with m degrees of freedom, L is a factor of the Cholesky decomposition $R = LL^T$ and $\rho^{x,v}$ is as introduced in Lemma 3.3.4.

Proof. Since $\xi \sim \mathcal{N}(0, R)$, the probability function φ gets the representation (3.3.5). With $g(\bar{x}, 0) > 0$, let U be a sufficiently small neighbourhood of \bar{x} such that for all $x \in U$ we still have that $g(x, 0) > 0$ and that the polynomial growth condition is satisfied at x . Then, the partial derivative $\nabla_x e$ of the function e defined in (3.3.6) exists on $U \times \mathbb{S}^{m-1}$ by Corollary 3.3.11 and is continuous there by Corollary 3.3.12. By compactness of \mathbb{S}^{m-1} , there exists some $K > 0$ such that

$$\|\nabla_x e(\bar{x}, v)\| \leq K \quad \forall v \in \mathbb{S}^{m-1}.$$

Again, continuity of $\nabla_x e$ on $U \times \mathbb{S}^{m-1}$ and compactness of \mathbb{S}^{m-1} guarantee that the function $\alpha : U \rightarrow \mathbb{R}$ defined by

$$\alpha(x) := \max_{v \in \mathbb{S}^{m-1}} \|\nabla_x e(x, v)\|$$

is continuous. Since $\alpha(\bar{x}) \leq K$, we may assume, after possibly shrinking U , that $\alpha(x) \leq 2K$ for all $x \in U$, whence

$$\|\nabla_x e(x, v)\| \leq 2K \quad \forall x \in U \quad \forall v \in \mathbb{S}^{m-1}. \quad (3.3.19)$$

From $\mu_\zeta(\mathbb{S}^{m-1}) = 1$ for the law μ_ζ of the uniform distribution on \mathbb{S}^{m-1} we infer that the constant $2K$ is an integrable function on \mathbb{S}^{m-1} uniformly dominating $\|\nabla_x e(x, v)\|$ on \mathbb{S}^{m-1} for all $x \in U$. Now, Lebesgue's dominated convergence theorem allows us to differentiate (3.3.5) under the integral sign:

$$\nabla \varphi(\bar{x}) = \int_{v \in \mathbb{S}^{m-1}} \nabla_x e(\bar{x}, v) d\mu_\zeta.$$

As stated in the beginning of this proof, the assumptions 1. and 2. imposed in the Theorem for the fixed point \bar{x} keep to hold for all x in the neighbourhood U . Therefore, we may derive that

$$\nabla \varphi(x) = \int_{v \in \mathbb{S}^{m-1}} \nabla_x e(x, v) d\mu_\zeta \quad \forall x \in U. \quad (3.3.20)$$

Exploiting once more the dominance argument from (3.3.19), the continuity of $\nabla_x e$ on $U \times \mathbb{S}^{m-1}$ and the compactness of \mathbb{S}^{m-1} ensure by virtue of Lebesgue's dominated convergence Theorem that $\nabla \varphi$ is continuous. Finally, formula (3.3.18) follows directly from Corollary 3.3.11. \square

Remark 3.3.14. *Evidently, formula (3.3.18) is explicit and can be used inside Deák's method in order to calculate $\nabla \varphi$ in parallel with φ by efficient sampling on \mathbb{S}^{m-1} . For each sampled point $v \in \mathbb{S}^{m-1}$ one first has to check whether the equation $g(x, rLv) = 0$ has a solution $r \geq 0$ at all. If not so ($v \in I(x)$), then such v does not contribute to the (approximated) integral in (3.3.18). Otherwise ($v \in F(x)$), one has to evaluate the integrand in (3.3.18) which amounts to finding the unique solution $r \geq 0$ of the equation $g(x, rLv) = 0$. In general, a few Newton-Raphson iterations should do the job.*

We now want to focus our attention on the assumptions of Theorem 3.3.13. First, recall that assuming a standard Gaussian distribution $\xi \sim \mathcal{N}(0, R)$ does not mean any loss of generality by virtue of Remark 3.3.2. Also assumption 1. of the Theorem is not restrictive. This will come as a consequence of the following proposition:

Proposition 3.3.15. *With g and φ as in Theorem 3.3.13, let the following assumptions be satisfied at some \bar{x} :*

1. *There exists some \bar{z} such that $g(\bar{x}, \bar{z}) > 0$.*

2. $\varphi(\bar{x}) > 1/2$.

Then, $g(\bar{x}, 0) > 0$.

Proof. As in the proof of Theorem 3.3.13 we may assume that $\xi \sim \mathcal{N}(0, R)$ so that φ gets the representation (3.3.5). Define the set $M := \{z \in \mathbb{R}^m | g(\bar{x}, z) \geq 0\}$. Clearly, M is convex and nonempty by our assumption 1. This same assumption (Slater point) guarantees that

$$\text{int } M = \{z \in \mathbb{R}^m | g(\bar{x}, z) > 0\}.$$

Assume that $g(\bar{x}, 0) \leq 0$. Then $0 \notin \text{int } M$ and, hence, one could separate 0 from M , which would mean that there exists some $c \in \mathbb{R}^m \setminus \{0\}$ such that

$$M \subseteq \{z \in \mathbb{R}^m | c^T z \geq 0\} =: \tilde{M}.$$

With ξ having a centered Gaussian distribution, the one-dimensional random variable $c^T \xi$ has a centered Gaussian distribution too and, hence, we arrive with our assumption 3. at the contradiction

$$1/2 = \mathbb{P}[c^T \xi \geq 0] = \mathbb{P}[\xi \in \tilde{M}] \geq \mathbb{P}[\xi \in M] = \varphi(\bar{x}) > 1/2.$$

□

The proposition means that violation of the first assumption in Theorem 3.3.13 implies that $g(\bar{x}, z) \leq 0$ for all z or that $\varphi(\bar{x}) \leq 1/2$. A typical application of Theorem 3.3.13 is probabilistic programming where one is imposing the chance constraint $\varphi(x) \geq p$ with some probability level p close to one. Since gradients of φ are usually calculated at or close to feasible points (e.g. by cutting planes), the case $\varphi(\bar{x}) \leq 1/2$ is very unlikely to occur. On the other hand, $g(\bar{x}, z) \leq 0$ for all z is a degenerate situation meaning that there exists no Slater point for the concave function $g(\bar{x}, \cdot)$. In such situation it typically happens that the set $\{z | g(x, z) \geq 0\}$ becomes empty for x arbitrarily close to \bar{x} which would entail a discontinuity of φ at \bar{x} . Then, of course, there is no hope to calculate a gradient at all.

Finally, turning to condition 2. of Theorem 3.3.13 (growth condition) it may require some technical effort to check it in concrete applications (see, e.g., the examples discussed in the following section). On the other hand, we shall see in a moment that we may do without this condition in case that the set $\{z | g(\bar{x}, z) \geq 0\}$ is bounded. To formulate a corresponding statement we need the following two auxiliary results:

Lemma 3.3.16. *Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be continuous. Moreover, let g be convex in the second argument. Then, for any $x \in \mathbb{R}^n$ with $g(x, 0) > 0$ one has that $I(x) = \emptyset$ if and only if $M(x) := \{z \in \mathbb{R}^m | g(x, z) \geq 0\}$ is bounded.*

Proof. Let x be arbitrary such that $g(x, 0) > 0$. Obviously boundedness of $M(x)$ implies that $I(x) = \emptyset$, so let us assume that $I(x) \neq \emptyset$ and that $M(x)$ is unbounded. Then, there is a sequence z_n with $g(x, z_n) \geq 0$ and $\|z_n\| \rightarrow \infty$. Without loss of generality,

we may assume that $\|z_n\|^{-1} z_n \rightarrow z$ for some $z \in \mathbb{R}^m \setminus \{0\}$. Let $t \geq 0$ be arbitrary. Then, $\|z_n\|^{-1} t \leq 1$ for n sufficiently large. From concavity of $g(x, \cdot)$, $g(x, 0) > 0$ and $g(x, z_n) \geq 0$ we infer that $g(x, \|z_n\|^{-1} t z_n) \geq 0$ for n sufficiently large. Passing to the limit, we get that $g(x, tz) \geq 0$. Thus, as $t \geq 0$ was arbitrary,

$$g(x, tz) \geq 0 \quad \forall t \geq 0. \quad (3.3.21)$$

Assume that there was some $\tau \geq 0$ with $g(x, \tau z) = 0$. Then, again by concavity of $g(x, \cdot)$ and by $g(x, 0) > 0$, one would arrive at the following contradiction with (3.3.21):

$$g(x, tz) < g(x, \tau z) = 0 \quad \forall t > \tau.$$

Hence, actually $g(x, tz) > 0$ for all $t \geq 0$. Putting $v := L^{-1}z / \|L^{-1}z\|$ - where L is the (invertible) matrix appearing in the definition of $I(x)$ - and observing that this definition is correct due to $z \neq 0$, we derive that $g(x, t\|L^{-1}z\|Lv) > 0$ for all $t \geq 0$. Since $\|L^{-1}z\| > 0$, this implies that $g(x, rLv) > 0$ for all $r \geq 0$. Hence the contradiction $v \in I(x)$ with our assumption $I(x) = \emptyset$. It follows that $M(x)$ is bounded as was to be shown. \square

Proposition 3.3.17. *Let g be as in Lemma 3.3.16 and $\bar{x} \in \mathbb{R}^n$ with $g(\bar{x}, 0) > 0$. If $M(\bar{x})$ is bounded, then there is a neighbourhood U of \bar{x} such that $M(x)$ remains bounded for all $x \in U$.*

Proof. By continuity of g , we may choose U small enough that $g(x, 0) > 0$ for all $x \in U$. If the assertion was not true, then by virtue of Lemma 3.3.16 there exists a sequence $x_n \rightarrow \bar{x}$ such that $I(x_n) \neq \emptyset$ for all $n \in \mathbb{N}$. By 1. in Lemma 3.3.3 this implies the existence of another sequence $v_n \in \mathbb{S}^{m-1}$ such that

$$g(x_n, rLv_n) > 0 \quad \forall r \geq 0 \quad \forall n \in \mathbb{N}.$$

Without loss of generality, we may assume that $v_n \rightarrow \bar{v}$ for some $\bar{v} \in \mathbb{S}^{m-1}$. For each $r \geq 0$ we may pass to the limit in the relation above, in order to derive that $g(\bar{x}, rL\bar{v}) \geq 0$ for all $r \geq 0$. With the same reasoning as below (3.3.21) we may conclude that indeed $g(\bar{x}, rL\bar{v}) > 0$ for all $r \geq 0$. This means that $\bar{v} \in I(\bar{x})$, whence $M(\bar{x})$ is unbounded by Lemma 3.3.16. This is a contradiction with our assumption. \square

Now we are in a position to state an alternative variant of Theorem 3.3.13 which does not require the verification of the growth condition:

Theorem 3.3.18. *Theorem 3.3.13 remains true if the second condition (growth condition) is replaced by the condition that the set $\{z | g(\bar{x}, z) \geq 0\}$ is bounded. Then, (3.3.18) becomes*

$$\nabla \varphi(x) = - \int_{v \in \mathbb{S}^{m-1}} \frac{\chi(\rho^{x,v}(x, v))}{\langle \nabla_z g(x, \rho^{x,v}(x, v) Lv), Lv \rangle} \nabla_x g(x, \rho^{x,v}(x, v) Lv) d\mu_\zeta(v) \quad \forall x \in U. \quad (3.3.22)$$

Proof. As in the proof of Theorem 3.3.13, the function e is continuous on $U \times \mathbb{S}^{m-1}$ by Corollary 3.3.6 because this result does not require the growth condition to hold. Moreover, $\nabla_x e$ exists on $U \times \mathbb{S}^{m-1}$. Indeed, our boundedness assumption ensures via Proposition 3.3.17 that - after possibly shrinking the neighbourhood U of \bar{x} - the set $\{z | g(x, z) \geq 0\}$ remains bounded for all $x \in U$. Lemma 3.3.16 implies that $I(x) = \emptyset$ or, equivalently according to 2. in Lemma 3.3.3 - that $F(x) = \mathbb{S}^{m-1}$ for all $x \in U$. Then, Corollary 3.3.7 yields that $\nabla_x e$ exists on $U \times \mathbb{S}^{m-1}$ and is given by

$$\nabla_x e(x, v) = - \frac{\chi(\rho^{x,v}(x, v))}{\langle \nabla_z g(x, \rho^{x,v}(x, v) Lv), Lv \rangle} \nabla_x g(x, \rho^{x,v}(x, v) Lv).$$

Since all occurring functions are continuous, the same holds true for $\nabla_x e$. Now the same argument as in the proof of Theorem 3.3.13 allows us to derive (3.3.20) which along with the formula for $\nabla_x e$ above yields (3.3.22). \square

3.3.2 Selected Examples

In this section we are going to discuss some instances of the probabilistic constraint (3.3.1) to which our gradient formulae obtained in Theorems 3.3.13 and 3.3.18 apply and thus could be used in the numerical solution of corresponding optimization problems.

3.3.2.1 Gaussian distributions

We assume first, as before, that the random vector has a Gaussian distribution. We shall focus on the particular model

$$\mathbb{P}[\langle f(\xi), h_1(x) \rangle \leq h_2(x)] \geq p \quad (3.3.23)$$

with nonlinear mappings $f : \mathbb{R}^m \rightarrow \mathbb{R}^l$ and $h_1 : \mathbb{R}^n \rightarrow \mathbb{R}^l$ and $h_2 : \mathbb{R}^n \rightarrow \mathbb{R}$ involving a coupling of random and decision vector.

Proposition 3.3.19. *In the probabilistic constraint (3.3.23), let f, h_1, h_2 be continuously differentiable, let the components f_i of f be convex and the components $h_{1,i}$ of h_1 be nonnegative. Furthermore, let $\xi \sim \mathcal{N}(0, R)$ have a standard Gaussian distribution with correlation matrix R and associated Cholesky decomposition $R = LL^T$. Consider any \bar{x} with $\langle f(0), h_1(\bar{x}) \rangle < h_2(\bar{x})$. Finally, let f satisfy the following polynomial growth condition:*

$$\|f(z)\| \leq \|z\|^\varkappa \quad \forall z : \|z\| \geq C$$

for certain $\varkappa, C > 0$. Then the probability function $\varphi(x) := \mathbb{P}[\langle f(\xi), h_1(x) \rangle \leq h_2(x)]$ defining the constraint (3.3.23) is continuously differentiable on a neighbourhood U of \bar{x} and its gradient is given by

$$\begin{aligned} \nabla \varphi(x) &= \int_{v \in F(x)} \chi(\rho^{x,v}(x, v)) [\langle h_1^T(x) \nabla f(\rho^{x,v}(x, v) Lv), Lv \rangle]^{-1} \nabla h_2(x) d\mu_\zeta(v) \\ &- \int_{v \in F(x)} \frac{\chi(\rho^{x,v}(x, v)) [f(\rho^{x,v}(x, v) Lv)]^T \nabla h_1(x)}{\langle h_1^T(x) \nabla f(\rho^{x,v}(x, v) Lv), Lv \rangle} d\mu_\zeta(v) \quad \forall x \in U. \end{aligned}$$

Proof. In our setting the general function g in (3.3.4) becomes $g(x, z) = -\langle f(z), h_1(x) \rangle + h_2(x)$. The continuous differentiability and concavity with respect to the second argument of g are evident from our assumptions. Moreover, $g(\bar{x}, 0) > 0$. As for the growth condition, let U be a neighbourhood of \bar{x} on which $\max\{\|\nabla h_1\|, \|\nabla h_2\|\} \leq K$ for some $K > 0$. Then, taking without loss of generality, the maximum norm, we have that

$$\begin{aligned} \|\nabla_x g(x, z)\| &= \left\| \nabla h_2(x) - [f(z)]^T \nabla h_1(x) \right\| \leq K(\|f(z)\| + 1) \\ &\leq \|z\|^{2+\varepsilon} \quad \forall x \in U, z : \|z\| \geq \max\{C, K, 2\}. \end{aligned}$$

Consequently, we may apply Theorem 3.3.13. (3.3.24) follows immediately from (3.3.18) for the given form of the function g . \square

3.3.3 Gaussian-like distributions

We are now going to apply Theorem 3.3.13 to probabilistic constraints with random vectors having non-Gaussian distributions. In a first case, we consider a linear probabilistic constraint

$$\mathbb{P}[\langle \eta, x \rangle \leq b] \geq p \tag{3.3.24}$$

with a random vector η whose components η_i ($i = 1, \dots, l$) are independent and have a χ^2 -distribution with n_i degrees of freedom. By definition, $\eta_i = \sum_{k=1}^{n_i} \xi_{i,k}^2$, where the $\xi_{i,k} \sim \mathcal{N}(0, 1)$ are independent for $k = 1, \dots, n_i$. We are interested in the gradient of the probability function $\varphi(x) := \mathbb{P}[\langle \eta, x \rangle \leq b]$. Define a Gaussian random vector with independent components

$$\xi := (\xi_{1,1}, \dots, \xi_{1,n_1}, \dots, \xi_{l,1}, \dots, \xi_{l,n_l}) \sim \mathcal{N}(0, I).$$

Clearly, $\eta \sim f(\xi)$, where $f_i(z) := \sum_{k=1}^{n_i} z_{i,k}^2$ for $i = 1, \dots, l$ and z is partitioned in the same way as ξ above. Then, the probability function defining (3.3.24) becomes

$$\varphi(x) = \mathbb{P}[\langle \eta, x \rangle \leq b] = \mathbb{P}[\langle f(\xi), x \rangle \leq b].$$

We derive the following gradient formula which does not need the verification of a polynomial growth condition and which is even fully explicit with respect to the resolving function $\rho^{x,v}$:

Proposition 3.3.20. *In (3.3.24), let $b > 0$. Consider any feasible point \bar{x} of (3.3.24) satisfying $\bar{x}_i > 0$ for $i = 1, \dots, n$. Then the probability function φ is continuously differentiable on a neighbourhood U of \bar{x} and its gradient is given by*

$$\nabla \varphi(x) = -\frac{\sqrt{b}}{2} \int_{v \in \mathbb{S}^{m-1}} \frac{\chi\left(\sqrt{b/\langle f(v), x \rangle}\right)}{\langle f(v), x \rangle^{3/2}} [f(v)]^T d\mu_\zeta(v) \quad \forall x \in U. \tag{3.3.25}$$

Proof. In our setting the general function g in (3.3.4) becomes $g(x, z) = -\langle f(z), x \rangle + b$ which is continuously differentiable. Since the components f_i are convex, $g(x, \cdot)$ is

concave whenever $x \geq 0$, which by our assumption holds true in a neighbourhood of \bar{x} . Evidently, the result of Theorems 3.3.13 and 3.3.18 are of local nature (differentiability around \bar{x}) so they actually do not need concavity of $g(x, \cdot)$ for all $x \in \mathbb{R}^n$ but only for x in a neighbourhood of \bar{x} which is satisfied here. Next observe that $g(\bar{x}, 0) = b > 0$. Finally, recalling that $\bar{x}_i > 0$ for $i = 1, \dots, n$, we obtain the estimate

$$\begin{aligned} \{z | g(\bar{x}, z) \geq 0\} &= \{z | \langle f(z), \bar{x} \rangle \leq b\} \subseteq \left\{ z \mid \left(\min_{i=1, \dots, n} \bar{x}_i \right) \sum_{i=1}^n f_i(z) \leq b \right\} \\ &= \left\{ z \mid \|z\|^2 \leq b \left(\min_{i=1, \dots, n} \bar{x}_i \right)^{-1} \right\}, \end{aligned}$$

whence the set on the left-hand side is bounded. Altogether, this allows us to invoke Theorem 3.3.18 and to derive the validity of formula (3.3.22). We now specify this formula in our setting. First observe that given $\xi \sim \mathcal{N}(0, I)$, we have that $R = I$, hence we have $L = I$ for the Cholesky decomposition $R = LL^T$. Next we calculate explicitly the function $\rho^{x,v}(x, v)$ which is the unique solution in $r \geq 0$ of the equation $\langle f(rLv), x \rangle = b$. Now, by definition of f ,

$$\langle f(rLv), x \rangle = r^2 \langle f(v), x \rangle = b,$$

whence

$$r = \sqrt{b / \langle f(v), x \rangle}. \quad (3.3.26)$$

Next, we calculate

$$\begin{aligned} -\nabla_x g(x, \rho^{x,v}(x, v) Lv) &= [f(\rho^{x,v}(x, v) v)]^T \\ &= [\rho^{x,v}(x, v)]^2 [f(v)]^T = (b / \langle f(v), x \rangle) [f(v)]^T \\ -\langle \nabla_z g(x, \rho^{x,v}(x, v) Lv), Lv \rangle &= \langle -\nabla_z g(x, \rho^{x,v}(x, v) v), v \rangle \\ &= \left\langle \sum_{i=1}^n x_i \nabla f_i(\rho^{x,v}(x, v) v), v \right\rangle \\ &= \sum_{i=1}^n x_i \langle \nabla f_i(\rho^{x,v}(x, v) v), v \rangle \\ &= 2\rho^{x,v}(x, v) \sum_{i=1}^n x_i \sum_{k=1}^{n_i} v_{i,k}^2 \\ &= 2\rho^{x,v}(x, v) \langle f(v), x \rangle = 2\sqrt{b \langle f(v), x \rangle}. \end{aligned}$$

Combination of these last relations with (3.3.26) provides formula (3.3.25). \square

As a second instance for a non-Gaussian but Gaussian-like distribution, we consider the multivariate log-normal distribution. Recall, that a random vector η follows a multivariate lognormal distribution if the vector $\xi := \log \eta$ (componentwise logarithm) has a Gaussian distribution. We consider now a probabilistic constraint of type

$$\mathbb{P}[\langle \eta, x \rangle \leq h(x)] \geq p \quad (3.3.27)$$

where η is an m -dimensional random vector with lognormal distribution and $h : \mathbb{R}^m \rightarrow \mathbb{R}$ is some function. We are interested in the gradient of the associated probability function $\varphi(x) := \mathbb{P}[\langle \eta, x \rangle \leq h(x)]$. We denote by $\xi := \log \eta$ the Gaussian random vector associated with η . Without loss of generality (see Remark 3.3.2) we may assume that $\xi \sim \mathcal{N}(0, R)$ for some correlation matrix R . We denote by L the associated factor in the Cholesky decomposition $R = LL^T$.

Proposition 3.3.21. *In the setting above, assume that \bar{x} satisfies $\bar{x}_i > 0$ for $i = 1, \dots, m$. Assume moreover that h is continuously differentiable and that $h(\bar{x}) > \sum_{i=1}^m \bar{x}_i$. Then,*

$$\nabla \varphi(x) = - \int_{\{v \in \mathbb{S}^{m-1} | \exists i: L_i v > 0\}} \frac{\chi(\rho^{x,v}(x, v))}{\sum_{i=1}^m x_i e_i^{\rho^{x,v}(x,v) L_i v} L_i v} [e^{\rho^{x,v}(x,v) L v} - \nabla h(x)] d\mu_\zeta(v) \quad \forall x \in U.$$

Here, L_i refers to the i th row of L and the expression e^z has to be understood componentwise.

Proof. In our setting the general function g in (3.3.4) becomes $g(x, z) = \langle -e^z, x \rangle + h(x)$. Clearly, g is continuously differentiable and concave with respect to z for all x close to \bar{x} (as mentioned in the proof of Proposition (3.3.24) this weakened condition is enough in the context of Theorem 3.3.13). Moreover, $g(\bar{x}, 0) = -\sum_{i=1}^m \bar{x}_i + h(\bar{x}) > 0$. In order to apply Theorem 3.3.13, it is sufficient to verify the exponential growth condition of Remark 3.3.10 (note that the originally imposed polynomial growth condition would not hold true here). To this aim, let U be a neighbourhood of \bar{x} on which $\|\nabla h\| \leq K$ for some $K > 0$. Then, with respect to the maximum norm, we get that

$$\|\nabla_x g(x', z)\| \leq \|e^z\| + \|\nabla h(x')\| \leq e^{\|z\|} + K \leq 2e^{\|z\|} \quad \forall x' \in U(x) \quad \forall z : \|z\| \geq \log K.$$

Hence, the exponential growth condition of Remark 3.3.10 is satisfied. This allows us to apply Theorem 3.3.13. Inserting the corresponding derivative formulae for g , we derive that φ is continuously differentiable on a neighbourhood U of \bar{x} and its gradient is given by

$$\nabla \varphi(x) = - \int_{v \in F(x)} \frac{\chi(\rho^{x,v}(x, v))}{\sum_{i=1}^m x_i e_i^{\rho^{x,v}(x,v) \langle L_i, v \rangle} \langle L_i, v \rangle} [e^{\rho^{x,v}(x,v) L v} - \nabla h(x)] d\mu_\zeta(v) \quad \forall x \in U. \quad (3.3.28)$$

Here, L_i denotes the i th row of the Cholesky factor L . To complete the proof, we have to verify the representation of the integration domain $F(x)$ asserted in the statement of this proposition. Without loss of generality, we assume the neighbourhood U of \bar{x} in the formula above to be small enough that $g(x, 0) > 0$ and $x_i > 0$ for $i = 1, \dots, m$ and for all $x \in U$ (recall that $g(\bar{x}, 0) > 0$ and $\bar{x}_i > 0$ for $i = 1, \dots, m$). We claim that for all $x \in U$ the set $I(x)$ introduced below (3.3.6) can be written as

$$I(x) = \{v \in \mathbb{S}^{m-1} | Lv \leq 0\}. \quad (3.3.29)$$

Indeed, let $x \in U$ and $v \in \mathbb{S}^{m-1}$ with $Lv \leq 0$ be arbitrary. Then, for all $r > 0$,

$$g(x, rLv) = -\langle e^{rLv}, x \rangle + h(x) \geq -\langle e^0, x \rangle + h(x) = g(x, 0) > 0,$$

whence $v \in I(x)$ by 1. of Lemma 3.3.3. Conversely, let $x \in U$ and $v \in I(x)$ be arbitrary. Then, $\langle e^{rLv}, x \rangle < h(x)$ for all $r > 0$. Define $J := \{i | L_i v > 0\}$. It follows from $x_i > 0$ for $i = 1, \dots, m$ that

$$h(x) > \sum_{i \in J} x_i e^{r \langle L_i, v \rangle}.$$

If $J \neq \emptyset$, then the sum on the right-hand side tends to ∞ for $r \rightarrow \infty$ which is a contradiction to this sum being bounded from above by $h(x)$ for all $r > 0$. Consequently, $J = \emptyset$, proving $Lv \leq 0$ and, thus, the reverse inclusion of (3.3.29). Since, by definition, $F(x) = \mathbb{S}^{m-1} \setminus I(x)$, we may plug the information from (3.3.29) into (3.3.28) in order to derive our asserted formula. \square

3.3.4 Student (or T-) distribution

As a last application, we are going to consider probabilistic constraints of type (3.3.1), where the random vector ξ follows a so-called multivariate Student or T- distribution. This is an important type of distribution in particular due to its application in the context of copulas. We recall that $\xi \sim \mathcal{T}(\mu, \Sigma, \nu)$ - i.e., ξ obeys a multivariate T-distribution with parameters μ, Σ, ν - if $\xi = \mu + \vartheta \sqrt{\frac{\vartheta}{u}}$, where $\vartheta \sim \mathcal{N}(0, \Sigma)$ has a multivariate Gaussian distribution with mean μ and covariance matrix Σ , $u \sim \chi^2(\nu)$ has a chi-squared distribution with ν degrees of freedom and ϑ and u are independent [156]. We are interested in the probability function (3.3.4) but this time for a T-variable rather than for a Gaussian one.

Remark 3.3.22. *Using the definition of a T- distribution, we may copy the arguments of Remark 3.3.2 in order to convince ourselves that in the consideration of (3.3.4) we may assume without loss of generality that $\xi \sim \mathcal{T}(0, R, \nu)$, where R is a correlation matrix. In particular, this can be arranged without disturbing the assumption of g in (3.3.4) being continuously differentiable and convex with respect to the second argument.*

In a first step, we provide an expression for the probability function (3.3.4) in case of a T-distribution:

Theorem 3.3.23. *Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a continuously differentiable function which is concave with respect to the second argument. Moreover, let $\xi \sim \mathcal{T}(0, R, \nu)$ for some correlation matrix R . Consider a point \bar{x} such that $g(\bar{x}, 0) > 0$. Then, there exists a neighbourhood U of \bar{x} such that the probability function (3.3.4) admits the representation*

$$\varphi(x) = \int_{v \in \mathbb{S}^{m-1}} \tilde{e}(x, v) d\mu_\zeta \quad \forall x \in U,$$

where for all $x \in U$ and $v \in \mathbb{S}^{m-1}$

$$\tilde{e}(x, v) := \begin{cases} F_{m, \nu}(m^{-1} [\rho^{x, v}(x, v)]^2) & v \in F(x) \\ 1 & v \in I(x) \end{cases}$$

and $F_{m,\nu}$ refers to the distribution function of the Fisher-Snedecor distribution with m and ν degrees of freedom. Moreover, $\rho^{x,v}$ is as introduced in Lemma 3.3.4 and $F(x)$ and $I(x)$ are defined in Lemma 3.3.3.

Proof. Let U be a neighbourhood of \bar{x} small enough such that $g(x, 0) > 0$ for all $x \in U$. Fix an arbitrary $x \in U$. According to the definition of ξ , there exist $\vartheta \sim \mathcal{N}(0, R)$ and $u \sim \chi^2(\nu)$ such that ϑ and u are independent and

$$\varphi(x) = \mathbb{P} \left[g(x, \vartheta \sqrt{\frac{\nu}{u}}) \geq 0 \right] = \int_{\{(y,t)|t>0, g(x, y\sqrt{\frac{\nu}{t}}) \geq 0\}} f_{\vartheta, u}(y, t) dy dt,$$

where $f_{\vartheta, u}$ denotes the joint density of the vector (ϑ, u) . By independence, $f_{\vartheta, u}(y, t) = f_{\vartheta}(y) f_u(t)$ where f_{ϑ} and f_u are the densities of ϑ and u , respectively. In particular, with Γ referring to the Gamma function, it holds that

$$f_u(t) = \begin{cases} \frac{1}{2^{\nu/2} \Gamma(\nu/2)} t^{\nu/2-1} e^{-t/2} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (3.3.30)$$

Therefore,

$$\begin{aligned} \varphi(x) &= \int_0^\infty \left(\int_{\{y|g(x, y\sqrt{\frac{\nu}{t}}) \geq 0\}} f_{\vartheta}(y) dy \right) f_u(t) dt \\ &= \frac{1}{2^{\nu/2} \Gamma(\nu/2)} \int_0^\infty \mathbb{P} \left[g(x, \vartheta \sqrt{\frac{\nu}{t}}) \geq 0 \right] t^{\nu/2-1} e^{-t/2} dt. \end{aligned} \quad (3.3.31)$$

With $M := \{z \in \mathbb{R}^m | g(x, z) \geq 0\}$ one has that, for $t > 0$,

$$\mathbb{P} \left[g(x, \vartheta \sqrt{\frac{\nu}{t}}) \geq 0 \right] = \mathbb{P} \left[\vartheta \in \frac{t}{\sqrt{\nu}} M \right].$$

Since $\vartheta \sim \mathcal{N}(0, R)$, (3.3.2) yields that for all $t > 0$

$$\begin{aligned} \mathbb{P} \left[g(x, \vartheta \sqrt{\frac{\nu}{t}}) \geq 0 \right] &= \int_{v \in \mathbb{S}^{m-1}} \mu_\eta \left(\{r \geq 0 | \frac{\sqrt{\nu}}{t} r L v \in M\} \right) d\mu_\zeta \\ &= \int_{v \in \mathbb{S}^{m-1}} \mu_\eta \left(\{r \geq 0 | g(x, \frac{\sqrt{\nu}}{t} r L v) \geq 0\} \right) d\mu_\zeta, \end{aligned}$$

where η has a χ -distribution with m degrees of freedom and ζ has a uniform distribution over \mathbb{S}^{m-1} . Moreover, L is a factor of the Cholesky decomposition $R = LL^T$. Let $t > 0$ be arbitrary. Assume first that $v \in F(x)$. With $g(x, 0) > 0$, let $\rho^{x,v} : \tilde{U} \times \tilde{V} \rightarrow \mathbb{R}_+$ be the function defined on certain neighbourhoods \tilde{U}, \tilde{V} of x and v , respectively. It follows from 1. in Lemma 3.3.4 that

$$\{r \geq 0 | g(x, \frac{\sqrt{\nu}}{t} r L v) \geq 0\} = \left[0, \frac{t}{\sqrt{\nu}} \rho^{x,v}(x, v) \right].$$

If in contrast $v \in I(x)$ then $g(x, rLv) > 0$ for all $r \geq 0$, whence

$$\{r \geq 0 | g(x, \frac{\sqrt{\nu}}{t} rLv) \geq 0\} = \mathbb{R}_+.$$

Combining this with (3.3.31), we conclude that

$$\begin{aligned} \varphi(x) &= \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty \int_{v \in F(x)} \mu_\eta \left(\left[0, \frac{t}{\sqrt{\nu}} \rho^{x,v}(x, v) \right] \right) d\mu_\zeta t^{\nu/2-1} e^{-t/2} dt \\ &+ \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty \int_{v \in I(x)} \mu_\eta(\mathbb{R}_+) d\mu_\zeta t^{\nu/2-1} e^{-t/2} dt \\ &= \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty \left(\int_{v \in F(x)} F_\eta \left(\frac{t}{\sqrt{\nu}} \rho^{x,v}(x, v) \right) d\mu_\zeta + \mu_\zeta(I(x)) \right) t^{\nu/2-1} e^{-t/2} dt \\ &= \mu_\zeta(I(x)) + \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty t^{\nu/2-1} e^{-t/2} \int_{v \in F(x)} F_\eta \left(\frac{t}{\sqrt{\nu}} \rho^{x,v}(x, v) \right) d\mu_\zeta dt, \end{aligned} \quad (3.3.32)$$

where F_η denotes the distribution function of η and we exploited that $F_\eta(0) = 0$, $\mu_\eta(\mathbb{R}_+) = 1$ and

$$\frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty t^{\nu/2-1} e^{-t/2} dt = \int_{\mathbb{R}} f_u(t) dt = 1.$$

Now, let $r \geq 0$ be arbitrary and let ζ have a Fisher-Snedecor (F-) distribution with m and ν degrees of freedom. Then, $\zeta = (\nu U_m) / (m U_\nu)$, where U_m and U_ν are independent and follow χ^2 -squared distributions with m and ν degrees of freedom, respectively. Denoting by $F_{m,\nu}$ the distribution function of ζ , we derive that

$$F_{m,\nu}(m^{-1}r^2) = \mathbb{P}(U_\nu^{-1}U_m \leq \nu^{-1}r^2) = \int_{\{(\tau,t) | \nu\tau \leq tr^2\}} f_{U_m, U_\nu}(\tau, t) d\tau dt,$$

where f_{U_m, U_ν} denotes the joint density of the vector (U_m, U_ν) . By independence, $f_{U_m, U_\nu}(\tau, t) = f_{U_m}(\tau) f_{U_\nu}(t)$ where the single χ^2 -densities are defined with appropriate degrees of freedom in (3.3.30). It follows that

$$\begin{aligned} F_{m,\nu}(m^{-1}r^2) &= \int_0^\infty \int_0^{tr^2/\nu} f_{U_\nu}(t) f_{U_m}(\tau) d\tau dt \\ &= \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty t^{\nu/2-1} e^{-t/2} \frac{1}{2^{m/2}\Gamma(m/2)} \int_0^{tr^2/\nu} \tau^{m/2-1} e^{-\tau/2} d\tau dt \\ &= \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty t^{\nu/2-1} e^{-t/2} \frac{1}{2^{m/2-1}\Gamma(m/2)} \int_0^{r\sqrt{t/\nu}} s^{m-1} e^{-s^2/2} ds dt \\ &= \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty t^{\nu/2-1} e^{-t/2} \int_0^{r\sqrt{t/\nu}} f_\eta(s) ds dt. \end{aligned}$$

Here, we used that the variable η introduced above has a χ -distribution with m degrees of freedom and so its density is given by

$$f_\eta(s) = \begin{cases} \frac{1}{2^{m/2-1}\Gamma(m/2)} s^{m-1} e^{-s^2/2} & s \geq 0 \\ 0 & s < 0 \end{cases}.$$

Consequently, with F_η denoting the distribution function of η ,

$$\begin{aligned} F_{m,\nu}(m^{-1}r^2) &= \frac{1}{2^{\nu/2}\Gamma(\nu/2)} \int_0^\infty t^{\nu/2-1} e^{-t/2} F_\eta\left(r\sqrt{t/\nu}\right) dt \\ &= \frac{1}{2^{\nu/2-1}\Gamma(\nu/2)} \int_0^\infty s^{\nu-1} e^{-s^2/2} F_\eta\left(sr/\sqrt{\nu}\right) ds. \end{aligned} \quad (3.3.33)$$

Consequently, exploiting the definition \tilde{e} in the statement of Theorem 3.3.23, putting $r := \rho^{x,v}(x, v)$ in (3.3.33) and applying Fubini's theorem, we end up via (3.3.32) at

$$\begin{aligned} \int_{v \in \mathbb{S}^{m-1}} \tilde{e}(x, v) d\mu_\zeta &= \\ \mu_\zeta(I(x)) + \int_{v \in F(x)} F_{m,\nu}(m^{-1}[\rho^{x,v}(x, v)]^2) d\mu_\zeta &= \\ \mu_\zeta(I(x)) + \frac{1}{2^{\nu/2-1}\Gamma(\nu/2)} \int_0^\infty \int_{v \in F(x)} s^{\nu-1} e^{-s^2/2} F_\eta\left(s\rho^{x,v}(x, v)/\sqrt{\nu}\right) d\mu_\zeta ds &= \varphi(x). \end{aligned}$$

□

Now, we may copy the proof of Corollary 3.3.6 but with the function e there replaced by the function \tilde{e} introduced above and with the expression $F_\eta(\rho^{x,v}(x', v'))$ in statement 1. of Lemma 3.3.5 replaced by the expression $F_{m,\nu}(m^{-1}[\rho^{x,v}(x', v')]^2)$ in order to derive the continuity of \tilde{e} at any $x \in U$, where U is defined in the Theorem above. Next, we may copy the proof of Corollary 3.3.7 (again with the appropriate replacements) and get the following:

Corollary 3.3.24. *For any $x \in \mathbb{R}^n$ with $g(x, 0) > 0$ and $v \in F(x)$ the partial derivative w.r.t x of the function $\tilde{e} : \mathbb{R}^n \times \mathbb{S}^{m-1} \rightarrow \mathbb{R}$ defined in Theorem 3.3.23 exists and is given by*

$$\nabla_x \tilde{e}(x, v) = -2\rho^{x,v}(x, v) \frac{f_{m,\nu}(m^{-1}[\rho^{x,v}(x, v)]^2)}{m \langle \nabla_z g(x, \rho^{x,v}(x, v) Lv), Lv \rangle} \nabla_x g(x, \rho^{x,v}(x, v) Lv) \quad (3.3.34)$$

where

$$f_{m,\nu}(t) = \begin{cases} \frac{\Gamma(m/2+\nu/2)}{\Gamma(m/2)\Gamma(\nu/2)} m^{m/2} \nu^{\nu/2} t^{m/2-1} (mt + \nu)^{-(m+\nu)/2} & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (3.3.35)$$

is the density of the Fisher-Snedecor distribution with m and ν degrees of freedom, $\rho^{x,v}$ refers to the function introduced in Lemma 3.3.4 and L is a factor of the Cholesky decomposition $R = LL^T$.

It is the equivalent of Lemma 3.3.9 that requires some additional conditions and work:

Lemma 3.3.25. *Let x be such that $g(x, 0) > 0$ and that g satisfies the polynomial growth condition at x with coefficient $\varkappa < \nu$ (Def. 3.3.8). Consider any sequence $(x_k, v_k) \rightarrow (x, v)$ for some $v \in I(x)$ such that $v_k \in F(x_k)$. Then,*

$$\lim_{k \rightarrow \infty} \nabla_x \tilde{e}(x_k, v_k) = 0.$$

Proof. First observe that $\rho^{x_k, v_k}(x_k, v_k) \rightarrow \infty$ by 2. in Lemma 3.3.5. The arguments of Lemma 3.3.9 allow us to deduce that for k sufficiently large the estimates (3.3.10) and (3.3.12) still hold. Using (3.3.35), we may combine Corollary 3.3.24 with (3.3.10) and (3.3.12) in order to derive that

$$\begin{aligned} \|\nabla_x \tilde{e}(x_k, v_k)\| &= \\ &= \left\| \frac{2\rho^{x_k, v_k}(x_k, v_k) f_{m, \nu}(m^{-1} [\rho^{x, v}(x, v)]^2)}{m \langle \nabla_z g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k), Lv_k \rangle} \nabla_x g(x_k, \rho^{x_k, v_k}(x_k, v_k) Lv_k) \right\| \\ &\leq 2\nu^{\nu/2} \frac{\Gamma(m/2 + \nu/2)}{\Gamma(m/2)\Gamma(\nu/2)} \|L\|^\varkappa \delta_1^{-1} \rho^{x_k, v_k}(x_k, v_k)^{m+\varkappa} \left(1 + \frac{\rho^{x_k, v_k}(x_k, v_k)^2}{\nu}\right)^{-\frac{m+\nu}{2}} \rightarrow_k 0, \end{aligned}$$

where the last limit follows from $\rho^{x_k, v_k}(x_k, v_k) \rightarrow \infty$ and $\varkappa < \nu$. \square

Upon having established Lemma 3.3.25 the same arguments of Corollary 3.3.11 can be used to show that \tilde{e} is differentiable with respect to x and to derive a similar formula. This can be done since the proof of Corollary 3.3.11 uses only the properties of e and we have established the same properties for \tilde{e} . Accordingly, $\nabla_x \tilde{e}(x, v)$ is given by formula (3.3.34) if $v \in F(x)$ and $\nabla_x \tilde{e}(x, v) = 0$ if $v \in I(x)$. In the same way as in Corollary 3.3.12 one establishes the continuity of $\nabla_x \tilde{e}$ upon noting that $f_{m, \nu}(t)$ defined in (3.3.35) is also continuous. We thus arrive at the following key result, of which the proof is a verbatim copy of that of Theorem 3.3.13 (Again e and \tilde{e} have the same properties).

Theorem 3.3.26. *Let $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a continuously differentiable function which is concave with respect to the second argument. Consider the probability function φ defined in (3.3.4), where $\xi \sim \mathcal{T}(0, R, \nu)$. Let the following assumptions be satisfied at some \bar{x} :*

1. $g(\bar{x}, 0) > 0$.
2. g satisfies the polynomial growth condition at \bar{x} (Def. 3.3.8) with coefficient $\varkappa < \nu$.

Then, φ is continuously differentiable on a neighbourhood U of \bar{x} and it holds that

$$\nabla \varphi(x) = \int_{v \in F(x)} -\frac{2\rho^{x, v}(x, v) f_{m, \nu}(m^{-1} [\rho^{x, v}(x, v)]^2)}{m \langle \nabla_z g(x, \rho^{x, v}(x, v) Lv), Lv \rangle} \nabla_x g(x, \rho^{x, v}(x, v) Lv) d\mu_\zeta(v) \quad \forall x \in U. \quad (3.3.36)$$

Here, μ_ζ is the law of the uniform distribution over \mathbb{S}^{m-1} , $f_{m, \nu}$ is the density of the Fisher-Snedecor-distribution with m and ν degrees of freedom and $\rho^{x, v}$ is as introduced in Lemma 3.3.4.

In the above result, the degrees of freedom ν of $\xi \sim \mathcal{T}(0, R, \nu)$ imposes an important restriction on the growth condition. Hence, on the mappings g that can be allowed for. In Theorem 3.3.18 we were able to replace the growth condition by a boundedness assumption. This can also be done now. Again the proof of the following result is a verbatim copy of that of Theorem 3.3.18.

Theorem 3.3.27. *Theorem 3.3.26 remains true if the second condition (growth condition) is replaced by the condition that the set $\{z|g(\bar{x}, z) \geq 0\}$ is bounded. Then, (3.3.36) becomes:*

$$\nabla\varphi(x) = \int_{v \in \mathbb{S}^{m-1}} -\frac{2\rho^{x,v}(x,v)f_{m,\nu}(m^{-1}[\rho^{x,v}(x,v)]^2)}{m \langle \nabla_z g(x, \rho^{x,v}(x,v)Lv), Lv \rangle} \nabla_x g(x, \rho^{x,v}(x,v)Lv) d\mu_\zeta(v) \quad \forall x \in U. \quad (3.3.37)$$

Moreover this result holds for all $\nu \geq 1$.

Remark 3.3.28. *Theorem 3.3.27 in particular covers the case when ξ follows a multivariate Cauchy Distribution, i.e., $\xi \sim \mathcal{T}(0, R, 1)$. This case was excluded in Theorem 3.3.26.*

Example 3.3.29. *Let us consider the optimization problem*

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & c^\top x \\ \text{s.t.} \quad & \varphi(x) := \mathbb{P}[\eta_1^{x_1} \eta_2^{x_2} \leq 2] \geq 0.8, \\ & Ax \leq b \\ & x \geq 0 \end{aligned} \quad (3.3.38)$$

where $\eta = (\eta_1, \eta_2)$ follows a log-normal law with underlying multivariate Gaussian distribution having mean $(0, 0)$ and 2×2 correlation matrix R with coefficient $R_{12} = -0.88$. The cost vector c is defined as $c = (-4, -1)$. The matrix A is given by

$$A = \begin{pmatrix} -\frac{3}{2} & 1 \\ \frac{1}{2} & -1 \\ 1 & 1 \end{pmatrix},$$

$$b = \left(\frac{1}{2}, \frac{1}{2}, 4\right).$$

The unconstrained optimal solution is $x^* = (3, 1)$. This solution turns out to be infeasible since it only provides a probability level of 0.63 roughly. Upon defining the mapping $g : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ as $g(x, z) = -\exp(x_1 z_1 + x_2 z_2) + 2$, we can observe that the chance constraint $\varphi(x)$ is equivalent with $\varphi(x) = \mathbb{P}[g(x, \xi) \geq 0]$. Alternatively $\varphi(x)$ can be characterized as $\varphi(x) = \mathbb{P}[x_1 z_2 + x_2 z_1 \leq \log(2)]$ as the logarithm is a strictly increasing mapping. The last characterization allows us to deduce the convexity of the feasible set by Kataoka's result [121] whenever $p > \frac{1}{2}$ as in the example. The same result allows us to show that Problem (3.3.38) is in fact equivalent with

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & c^\top x \\ \text{s.t.} \quad & x^\top R x \leq \left(\frac{\log(r)}{\Phi^{-1}(0.8)}\right)^2, \\ & Ax \leq b \\ & x \geq 0 \end{aligned} \quad (3.3.39)$$

where Φ^{-1} denotes the inverse of the standard normal distribution function. The optimal solution of (3.3.39) is readily found to be $x^{\mathbf{x}} = (1.7250, 1.6010)$. One easily derives that $x^s = (0.2, 0.2)$ is a Slater point for problem (3.3.38).

Since the feasible set is convex, we can apply a Supporting Hyperplane method for solving (3.3.38) directly. Theorem 3.3.13 allows us to compute a gradient if the conditions are satisfied. Returning to our mapping g , it is easily seen that g is concave in z whenever we impose the additional constraint $x \geq 0$. Moreover $g(x, 0) > 0$ regardless of $x \geq 0$. One readily derives from the Cauchy-Schwarz inequality that $\|\nabla_x g(x, z)\| \leq \|z\| \exp(\|z\| \|x\|)$. Following Remark 3.3.10 this is also a sufficient condition for Lemma 3.3.9 to hold. This in turn applies that the polynomial growth condition of Theorem 3.3.13 can be substituted by the above exponential growth condition. We can thus use the formula (3.3.18) for computing a gradient of $\varphi(x)$. The numerical sampling scheme used for the integral over the sphere \mathbb{S}^{m-1} is the O_2 -estimator of Déak [50] where the uniform orthonormal system is generated according to the ideas of [55]. We have generated a 1000 orthonormal systems.

Let k be the current iteration of the supporting hyperplane algorithm and x_k the current candidate solution. We will first compute $\lambda > 0$ such that $x^c = \lambda x_k + (1 - \lambda)x^s$ is on the boundary of the feasible set, i.e., $\varphi(x^c) = p$. We then add the constraint $-\nabla\varphi(x^c)^\top x \leq -\nabla\varphi(x^c)^\top x^c$ to problem (3.3.38) and solve the newly obtained problem in order to obtain a new iterate. Each iterate x_k is, a priori, infeasible and provides a lower bound on the optimal value. Each iterate x^c is feasible and provides an upper bound. We will stop the algorithm if $\frac{c^\top x^c - c^\top x_k}{c^\top x_k} < 0.01$.

Figure (3.1) shows the results of the supporting hyperplane approach. The red ellipsoid is the feasible set for $\varphi(x) \geq 0.8$. The dashed red ellipsoid is the feasible set for $\varphi(x) \geq 0.79$, the red lines correspond to the feasible set of $Ax \leq b$ and the black star is the Slater point x^s . Successive iterates x_k are indicated by their number and a cross, the successive feasible points x_c are indicated by circles. Finally the green lines are the generated supporting hyperplanes and the diamond indicates the found optimal solution.

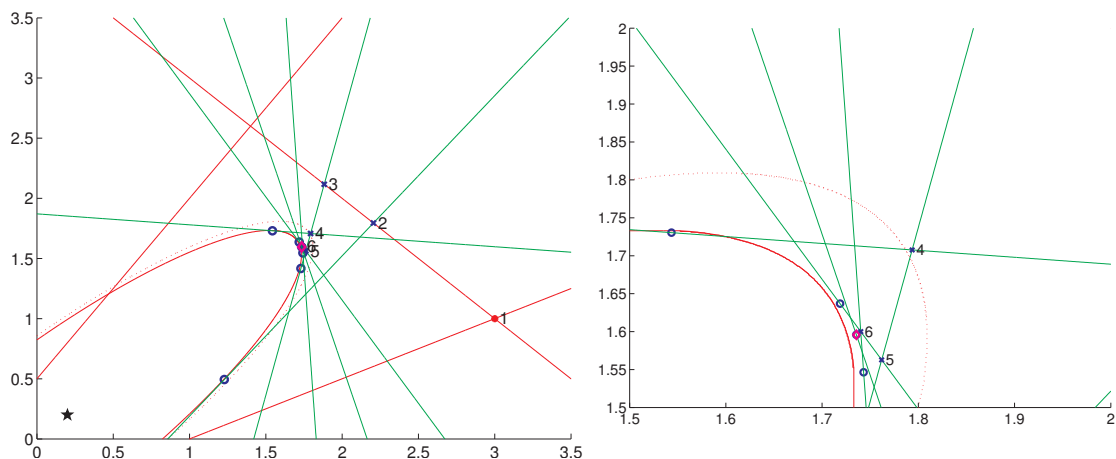


Figure 3.1: Illustration of solving problem (3.3.38) with the derived gradient formula. The right figure provides a zoom near the optimal solution.

The thus obtained solution is $x^* = (1.7356, 1.5959)$ with objective function value -8.5383 , whereas the truly optimal solution x^{\star} provides an objective function value of -8.5010 . This difference is explained by the fact that x^* is slightly infeasible due to numerical noise.

3.4 Eventual Convexity

Consider a constraint of the type (2.0.1), i.e.,

$$\mathbb{P}[g(x, \xi) \geq 0] \geq p, \quad (3.4.1)$$

where $g : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^k$ is a constraint mapping, $x \in \mathbb{R}^n$ the decision vector, $\xi \in \mathbb{R}^m$ a random variable, \mathbb{P} its associated probability measure and $p \in (0, 1)$ a pre-specified robustness-level.

A key question in Chance Constrained Programming is the convexity of the feasible set, i.e., of the set $M(p) = \{x \in \mathbb{R}^n : \mathbb{P}[g(x, \xi) \geq 0] \geq p\}$. It is well known ([51, 181, 182]) that if ξ admits a density with specific generalized concavity properties, and g is a (jointly) quasi-concave mapping, then indeed the feasible set is convex. In many practical applications, we have separable constraint mappings, i.e., $g(x, \xi) = h(x) - \xi$, where $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$. The requirement of joint quasi concavity of g then can be asserted if h is concave. Note that quasi concavity of h is not sufficient.

The latter requirement on the i -th component h_i of the mapping h can be relaxed to $-\alpha_i$ -concavity, $\alpha_i > 0$ if ξ has independent components and each component ξ_i has a so-called $\alpha_i + 1$ -decreasing density ([110]). This comes at the cost of only being able to assert convexity of $M(p)$ for p values larger than some threshold. Such convexity is called *eventual convexity* and is clearly sufficient in many practical applications when we are looking at large p values. Independence of components of ξ is a strong requirement, which was relaxed in a second work of the same authors [111]. Indeed, ξ can be allowed to have a dependence structure induced by a specially structured Copula, called a log-exp-concave Copula. Some well-known Copulae (Maximum, Independent, Gumbel) are log-exp-concave, as shown in [111]. However, it turns out that the Clayton Copula is not log-exp-concave.

When examining very carefully the results of [111], it appears that the link between generalized concavity of h , individual distribution functions F_i of each component ξ_i , $i = 1, \dots, m$ and the Copula is not clearly exhibited. As a result all log-exp concave Copulae actually provide the same probability threshold. Moreover this level depends in a way on one of the distribution functions F_i only. One can thus set up two versions of a problem wherein only one component of ξ has the same distribution and obtain convexity results for the same asserted probability threshold. In particular ξ can have independent components, or components linked through an arbitrary log-exp concave Copula and the same probability threshold is obtained. This is intuitively puzzling.

In this section, we will show that one can derive eventual convexity of the feasible set $M(p)$ for a larger class of Copulae. In particular, we will show that the Clayton Copula is in this extended class. We will moreover exhibit clearly the link between the generalized concavity properties of the mapping h , the individual distribution functions F_i , $i = 1, \dots, m$ and that of the Copula. We will also show that by adding some additional explicit constraints to the optimization problem, defining a convex feasible set, the probability threshold can be made to depend on the Copula. We will provide several examples showing that one can obtain eventual convexity results for lower p values than in the

papers [110, 111]. Finally we provide a characterization of the Gaussian Copula in the tail and show that it decomposes as the sum of log-exp concave functions.

This section is organized as follows. We will begin by introducing some useful notation and generalized concavity in section 3.4.1. We will define a class of Copulae containing the class of log-exp-concave Copula and characterize this class in section 3.4.2. In section 3.4.3 we will provide our main Theorems proving eventual convexity for the feasible sets $M(p)$ under specific conditions on the Copula, the individual distribution functions and constraint mappings h_i , $i = 1, \dots, m$. Section 3.4.4 provides a series of examples and results showing that sharper bounds on the probability threshold can be obtained and that this new class of Copulae contains strictly more Copulae than just log-exp-concave Copulae. In section 3.4.5 we derive a characterization of the Gaussian Copula, showing that it decomposes as the sum of log-concave functions, at least in its upper-tail. A potential application for modelling probabilistic Constraints with Copula is provided in section 3.4.6.

3.4.1 Notation

Throughout this section, we will apply many algebraic operations on vectors. In order to have short notation, these are understood componentwise. As an example, for any $u \in \mathbb{R}^m$, e^u will be defined as $e^u = (e^{u_1}, \dots, e^{u_m})$. In a very similar way, we will define $u^{\frac{1}{\gamma}}$ for $\gamma \neq 0$. For a mapping $h : \mathbb{R}^m \rightarrow \mathbb{R}$, $u \in \mathbb{R}^m \mapsto h(e^u)$ is thus understood as $u \mapsto h(e^{u_1}, \dots, e^{u_m})$. We will also extend this short notation to one-dimensional mappings applied to a vector. If $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ is a mapping, we mean $\varphi(u) = (\varphi(u_1), \dots, \varphi(u_m))$ when $u \in \mathbb{R}^m$.

Throughout this section, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ will be a constraint mapping, $\xi \in \mathbb{R}^m$ an m -dimensional random vector. The component ξ_i is assumed to have one dimensional distribution function $z \in \mathbb{R} \mapsto F_i(z) := \mathbb{P}[\xi_i \leq z]$, $i = 1, \dots, m$. Finally $C : [0, 1]^m \rightarrow [0, 1]$ is a Copula, such that

$$\mathbb{P}[\xi \leq h(x)] = C(F_1(h_1(x)), \dots, F_m(h_m(x))). \quad (3.4.2)$$

We will assume that the mapping (3.4.2) defines a constraint of a Stochastic optimization problem in the following way:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \mathbb{P}[\xi \leq h(x)] \geq p, \end{aligned} \quad (3.4.3)$$

for some probability level p and convex function f . This problem is assumed to be the "Stochastic" variant of the deterministic problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & b \leq h(x), \end{aligned} \quad (3.4.4)$$

for an appropriately chosen vector $b \in \mathbb{R}^m$, e.g., $b = \mathbb{E}(\xi)$. Assuming convexity of f , problem (3.4.4) is a convex optimization problem if and only if the mapping h has

some generalized concavity property. It is therefore natural to make such an assumption. Problem (3.4.3) arises, whenever problem (3.4.4) has turned out insufficiently robust and a decision vector x accounting for uncertainty is looked for. Such left-hand side uncertainty arises in many practical applications such as [246, 239].

In this section the set $M(p)$ will be defined as $M(p) := \{x \in \mathbb{R}^n : \mathbb{P}[\xi \leq h(x)] \geq p\}$, where $\mathbb{P}[\xi \leq h(x)]$ is as in equation (3.4.2).

3.4.2 Copulae and generalized concavity

In [111], the authors call a Copula C a log-exp concave Copula if and only if $u \in [0, 1]^m \mapsto \log(C(e^u))$ is a concave mapping. It therefore appears natural to provide the following extension of this concept. We will thus speak of a δ - γ -concave Copula.

Definition 3.4.1. *Let $\gamma \in \mathbb{R}$ be given, and let the set $X(\gamma)$ be defined as $X(\gamma) = [0, 1]^m$ for $\gamma > 0$, $X(0) = (-\infty, 0]^m$ and $X(\gamma) = [1, \infty)^m$ for $\gamma < 0$. Let $\delta \in [-\infty, \infty]$ be equally given. We call a Copula $C : [0, 1]^m \rightarrow [0, 1]$ δ - γ -concave if the mapping $u \in X(\gamma) \mapsto C(u^{\frac{1}{\gamma}})$ is δ -concave, whenever $\gamma \neq 0$ and $u \in X(0) \mapsto C(e^u)$ is δ -concave whenever $\gamma = 0$.*

The presence of the set $X(\gamma)$ is only to have the arguments mapped in $[0, 1]^m$, so that we can compose with the Copula afterwards.

Remark 3.4.2. *This is indeed an extension of the notion of log-exp-concave Copulae as defined in the paper [111]. Indeed a log-exp-concave Copula is 0-0-concave in our setting. Examples of log-exp-concave Copulae are the independent, maximum and Gumbel Copula. The latter is an Archimedean Copula, a family of Copulae generated by a one-dimensional function called the generator. We refer to [150] for a full characterization of generators of Archimedean Copulae.*

Remark 3.4.3. *Copulae are such that $C(u) = 0$ if and only if there is some $i = 1, \dots, m$ with $u_i = 0$. Pick $u \in [0, 1]^m$ with $C(u) = 0$, some $v \in [0, 1]^m$ and $\lambda \in [0, 1]$ and form $z = \lambda u + (1-\lambda)v$. Pick moreover an arbitrary $\gamma > 0$, it is then clear from C being a Copula that $C(z^{\frac{1}{\gamma}}) \geq 0$ and from the definition of δ -concavity that $m_\delta(C(u^{\frac{1}{\gamma}}), C(v^{\frac{1}{\gamma}}), \lambda) = 0$. It is therefore sufficient to verify δ - γ -concavity of a Copula with $\gamma > 0$ on $(0, 1]^m$ only. This avoids the problem of degenerate arguments. Degenerate arguments are naturally avoided when $\gamma \leq 0$.*

Remark 3.4.4. *One could alternatively replace the δ -concavity requirement with a g -concavity requirement as defined in [6] and [227]. We are however interested in eventual convexity, i.e., convexity of all levels sets above some threshold. The notion of g -concavity then implies quasi-concavity on that (sub)-set.*

Some Copulae might not be δ - γ -concave on the whole domain $X(\gamma)$ as the following example shows:

Example 3.4.5. Let $C : [0, 1]^m \rightarrow [0, 1]$ be the Gaussian Copula, i.e., $C(u) = \Phi^R(\varphi^{-1}(u))$, where Φ^R is the probability distribution function of a centered Gaussian random variable in dimension m with correlation matrix R and $\varphi^{-1} : (0, 1) \rightarrow \mathbb{R}$ the inverse of a standard normal probability distribution function in dimension 1. The mapping $\varphi^{-1} : (0, 1) \rightarrow \mathbb{R}$ is concave on $(0, \frac{1}{2}]$ and Φ^R is log-concave ([181]). It then follows that C is 0-1-concave on $[0, \frac{1}{2}]^m$. Indeed, pick $x, y \in [0, \frac{1}{2}]^m$, $\lambda \in [0, 1]$. From concavity of φ^{-1} and monotonicity of the distribution function we get

$$\begin{aligned} C(m_1(x, y, \lambda)) &\geq \Phi^R(m_1(\varphi^{-1}(x_1), \varphi^{-1}(y_1), \lambda), \dots, m_1(\varphi^{-1}(x_m), \varphi^{-1}(y_m), \lambda)) \\ &\geq m_0(C(x), C(y), \lambda). \end{aligned}$$

Remark 3.4.6. The above example does not exclude a more general δ - γ -concavity property on other sets.

This therefore motivates the following definition:

Definition 3.4.7. Let $q \in (0, 1)^m$ be some point and define the sets $X(q, \gamma)$ as follows $X(q, \gamma) = \prod_{i=1}^m [q_i^\gamma, 1]$ for $\gamma > 0$, $X(q, 0) = \prod_{i=1}^m [\log(q_i), 0]$ and $X(q, \gamma) = \prod_{i=1}^m [1, q_i^\gamma]$ for $\gamma < 0$. We call a Copula $C : [0, 1]^m \rightarrow [0, 1]$ δ - γ - q -concave if the mapping $u \in X(q, \gamma) \mapsto C(u^{\frac{1}{\gamma}})$ is δ -concave, whenever $\gamma \neq 0$ and $u \in X(q, 0) \mapsto C(e^u)$ is δ -concave whenever $\gamma = 0$.

Remark 3.4.8. The reason the sets are of the specific form in definition 3.4.7 and not their opposite will become apparent in Theorems 3.4.18, 3.4.19 and 3.4.20.

We will introduce one last notion, wherein δ - γ -concavity holds in some asymptotic way.

Definition 3.4.9. We call a Copula $C : [0, 1]^m \rightarrow [0, 1]$ asymptotically δ -concave if for each $\gamma > 0$ there exists a point $q(\gamma) \in (0, 1)^m$ such that C is δ - γ -concave when restricted to the set $\tilde{X}(\gamma) = [0, q(\gamma)]^m$ and moreover $\lim_{\gamma \downarrow 0} q(\gamma) = 1$.

3.4.2.1 Structure of the family of δ - γ -concave Copulae

At first sight it might appear that the family of δ - γ -concave Copulae is rather loose. In particular it may appear that the δ -0-Copulae fit in rather artificially. The following results show that this is not true and that the family is naturally ordered.

It follows from Lemma 2.5.5 that δ -concavity implies some "descending" order, i.e., a δ -concave mapping is also β -concave whenever $\beta \leq \delta$. It turns out that the effect of γ is ascending as the following lemma shows.

Lemma 3.4.10. Let $C : [0, 1]^m \rightarrow [0, 1]$ be a δ - β -concave Copula and let $\alpha \in \mathbb{R}$ be given such that $\beta \leq \alpha$. Then C is also δ - α -concave.

Proof. Pick any $x, y \in [0, 1]^m$ and $\lambda \in [0, 1]$ arbitrarily. We begin with the special case $\beta = 0$, and $\alpha > 0$. We derive from concavity of the log-function and monotonicity of the

exp function that:

$$\begin{aligned} m_1(x, y, \lambda)^{\frac{1}{\alpha}} &= \exp\left(\frac{1}{\alpha} \log m_1(x, y, \lambda)\right) \geq \exp\left(\frac{1}{\alpha} m_1(\log x, \log y, \lambda)\right) \\ &= \exp\left(m_1\left(\frac{1}{\alpha} \log x, \frac{1}{\alpha} \log y, \lambda\right)\right) = \exp\left(m_1\left(\log x^{\frac{1}{\alpha}}, \log y^{\frac{1}{\alpha}}, \lambda\right)\right), \end{aligned}$$

where with $x \in (0, 1]^m$, it follows that $\log(x^{\frac{1}{\alpha}}) \in X(0)$. Now from monotonicity of the Copula we get

$$\begin{aligned} C(m_1(x, y, \lambda)^{\frac{1}{\alpha}}) &\geq C(\exp(m_1(\log x^{\frac{1}{\alpha}}, \log y^{\frac{1}{\alpha}}, \lambda))) \\ &\geq m_\delta(C(x^{\frac{1}{\alpha}}, C(y^{\frac{1}{\alpha}}), \lambda), \end{aligned}$$

which was to be shown by δ -concavity of $u \mapsto C(e^u)$ and Remark 3.4.3. Now picking any $\beta \neq 0$ and $\alpha \geq \beta$, we define $z = x^{\frac{1}{\alpha}}$, $w = y^{\frac{1}{\alpha}}$, when $\alpha \neq 0$ and $z = \exp(x)$, $w = \exp(y)$ when $\alpha = 0$. Together with $x, y \in X(\alpha)$, this implies $z, w \in [0, 1]^m$. Now with $\beta \neq 0$, one obtains $z^\beta, w^\beta \in X(\beta)$. We then observe that $m_1(x, y, \lambda)^{\frac{1}{\alpha}} = m_\alpha(z, w, \lambda)$, when $\alpha \neq 0$ and $\exp(m_1(x, y, \lambda)) = m_0(z, w, \lambda)$ when $\alpha = 0$. It therefore follows from monotonicity of the Copula combined with Lemma 2.5.5 and δ - β -concavity of the Copula that

$$\begin{aligned} C(m_1(x, y, \lambda)^{\frac{1}{\alpha}}) &= C(m_\alpha(z, w, \lambda)) \geq C(m_\beta(z, w, \lambda)) \\ &\geq m_\delta(C(z), C(w), \lambda) = m_\delta(C(x^{\frac{1}{\alpha}}), C(y^{\frac{1}{\alpha}}), \lambda), \end{aligned}$$

when $\alpha \neq 0$ and for $\alpha = 0$, we get

$$\begin{aligned} C(\exp(m_1(x, y, \lambda))) &= C(m_0(z, w, \lambda)) \geq C(m_\beta(z, w, \lambda)) \\ &\geq m_\delta(C(z), C(w), \lambda) = m_\delta(C(e^x), C(e^y), \lambda), \end{aligned}$$

as was to be shown. □

Remark 3.4.11. *This is an extension of Proposition 3 of [111] where it is shown that 0-0-concavity implies 0-1-concavity.*

One can also prove this same lemma with the local δ - γ -concavity notion.

Lemma 3.4.12. *Let $q \in (0, 1)^m$ be given and let $C : [0, 1]^m \rightarrow [0, 1]$ be a δ - β - q -concave Copula and let α be given such that $0 \leq \beta \leq \alpha$. Then C is also δ - α - q -concave.*

Proof. The proof is identical to that of Lemma 3.4.10 except that we need to take care of the domains. To this end pick $x \in X(q, \alpha)$. We begin by considering the case $\beta = 0$, $\alpha > 0$. It then follows from monotonicity of the log function and of $x \mapsto x^{\frac{1}{\alpha}}$, that $\log x^{\frac{1}{\alpha}} \geq \log(q)$. In a similar way it follows with $\alpha > \beta > 0$ that $x^{\frac{\beta}{\alpha}} \geq q$, so that β - q -concavity can be applied. □

Since δ -concavity of an arbitrary mapping implies weaker concavity properties for the same mapping, we trivially derive the following corollary:

Corollary 3.4.13. *Let $C : [0, 1]^m \rightarrow [0, 1]$ be a δ - γ -concave Copula and let $\alpha \geq \gamma$ and $\beta \leq \delta$ be given. Then C is also β - α -concave.*

3.4.2.2 Tools for deriving δ - γ -Concavity of Copulae

Copulae are in particular probability distribution functions, some of them admitting densities. The following result is therefore a trivial consequence of Theorem 4.15 [51]:

Lemma 3.4.14. *Let $C : [0, 1]^m \rightarrow [0, 1]$ be a Copula, admitting a density $c : [0, 1]^m \rightarrow [0, 1]$. If the density c is α -concave for some $\alpha \geq -\frac{1}{m}$, then C is a δ -1-concave Copula for $\delta = \frac{\alpha}{1+m\alpha}$.*

The following Corollary follows immediately from this result

Corollary 3.4.15. *Let $C : [0, 1]^m \rightarrow [0, 1]$ be a Copula, admitting a density $c : [0, 1]^m \rightarrow [0, 1]$. If the density c is such that $u \in (-\infty, 0]^m \mapsto c(\exp(u))$ is log-concave then the Copula is 0-0-concave.*

Proof. We begin by remarking that $h : (-\infty, 0]^m \rightarrow [0, 1]$, defined as $h(u) = C(\exp(u))$ is also a distribution function. Since C admits a density, the density of h is given by

$$\frac{\partial^m}{\partial u_1 \cdots \partial u_m} h = c(e^u) \prod_{i=1}^m e^{u_i}.$$

Now the logarithm of the latter function is $\log(c(e^u)) + \sum_{i=1}^m u_i$, i.e., the sum of concave functions, i.e., concave. It follows from Lemma 3.4.14 that h is log-concave, i.e., C is 0-0-concave. \square

3.4.2.3 Estimates with Copulae

Before moving to our main result, we first derive some useful auxiliary results.

Lemma 3.4.16. *Let $C : [0, 1]^m \rightarrow [0, 1]$ be a δ - γ -concave Copula and $\infty \geq \gamma_i \geq \gamma$, $i = 1, \dots, m$ any sequence of numbers. For any $x, y \in [0, 1]^m$, $\lambda \in [0, 1]$, the following inequality holds*

$$C(m_{\gamma_1}(x_1, y_1, \lambda), \dots, m_{\gamma_m}(x_m, y_m, \lambda)) \geq m_\delta(C(x_1, \dots, x_m), C(y_1, \dots, y_m), \lambda).$$

Proof. Let $\gamma_i \geq \gamma$ for all $i = 1, \dots, m$ be given and pick $x, y \in [0, 1]^m$ and $\lambda \in [0, 1]$ in an arbitrary way. We begin by noting that $m_{\gamma_i}(x_i, y_i, \lambda) \geq m_\gamma(x_i, y_i, \lambda)$ for any $i \in \{1, \dots, m\}$, since this mapping is non-decreasing by Lemma 2.5.5. Copulae are distribution functions and hence are increasing in increasing arguments, so we get

$$C(m_{\gamma_1}(x_1, y_1, \lambda), \dots, m_{\gamma_m}(x_m, y_m, \lambda)) \geq C(m_\gamma(x_1, y_1, \lambda), \dots, m_\gamma(x_m, y_m, \lambda)). \quad (3.4.5)$$

Now, if $\gamma \neq 0$, the right-hand side of (3.4.5) is equal to

$$C(m_\gamma(x_1, y_1, \lambda), \dots, m_\gamma(x_m, y_m, \lambda)) = C(m_1(x_1^\gamma, y_1^\gamma, \lambda)^{\frac{1}{\gamma}}, \dots, m_1(x_m^\gamma, y_m^\gamma, \lambda)^{\frac{1}{\gamma}}) \quad (3.4.6)$$

and in the case $\gamma = 0$, we get

$$\begin{aligned} C(m_0(x_1, y_1, \lambda), \dots, m_0(x_m, y_m, \lambda)) = \\ C(\exp(m_1(\log x_1, \log y_1, \lambda)), \dots, \exp(m_1(\log x_m, \log y_m, \lambda))). \end{aligned} \quad (3.4.7)$$

For $z, w \in (0, 1]$, we can also derive that

$$m_1(z^\gamma, w^\gamma, \lambda) \in [\min\{z^\gamma, w^\gamma\}, \max\{z^\gamma, w^\gamma\}],$$

when $\gamma \neq 0$ and $m_1(\log z, \log w, \lambda) \in [\min\{\log(z), \log(w)\}, \max\{\log(z), \log(w)\}]$. This shows that

$$(m_1(x_1^\gamma, y_1^\gamma, \lambda), \dots, m_1(x_m^\gamma, y_m^\gamma, \lambda)) \in X(\gamma),$$

when $\gamma \neq 0$ and

$$(m_1(\log x_1, \log y_1, \lambda), \dots, m_1(\log x_m, \log y_m, \lambda)) \in X(0).$$

Hence, since the mappings on the right-hand side of (3.4.6) and (3.4.7) are δ -concave we obtain the estimates

$$C(m_{\gamma_1}(x_1, y_1, \lambda), \dots, m_{\gamma_m}(x_m, y_m, \lambda)) \geq m_\delta(C(x_1, \dots, x_m), C(y_1, \dots, y_m), \lambda),$$

as was to be shown. \square

Lemma 3.4.17. *Let $q \in (0, 1)^m$ be a given point and let $C : [0, 1]^m \rightarrow [0, 1]$ be a δ - γ - q -concave Copula. Assume furthermore that $\infty \geq \gamma_i \geq \gamma$, $i = 1, \dots, m$ is any sequence of numbers. Then for any $x, y \in [q, 1]^m$ and $\lambda \in [0, 1]$ the following inequality holds:*

$$C(m_{\gamma_1}(x_1, y_1, \lambda), \dots, m_{\gamma_m}(x_m, y_m, \lambda)) \geq m_\delta(C(x_1, \dots, x_m), C(y_1, \dots, y_m), \lambda).$$

Proof. Let $\gamma_i \geq \gamma$ for all $i = 1, \dots, m$ be given and pick $x, y \in [q, 1]^m$ and $\lambda \in [0, 1]$ in an arbitrary way. Using the arguments of the proof of Lemma 3.4.16 we can derive equations (3.4.6) and (3.4.7). It remains to show that

$$(m_1(x_1^\gamma, y_1^\gamma, \lambda), \dots, m_1(x_m^\gamma, y_m^\gamma, \lambda)) \in X(q, \gamma),$$

when $\gamma \neq 0$ and

$$(m_1(\log x_1, \log y_1, \lambda), \dots, m_1(\log x_m, \log y_m, \lambda)) \in X(q, 0).$$

Then we can apply again the δ -concavity inequality to derive the final estimate of the Lemma. To this end, pick $z, w \in [q, 1]$ arbitrarily and consider the case $\gamma \geq 0$. From monotonicity of the mapping $u \in [0, 1] \mapsto u^\gamma$ (or $u \in (0, 1] \mapsto \log(u)$), we obtain for $u \geq q$ that $u^\gamma \geq q^\gamma$ (or $\log(u) \geq \log(q)$). Altogether this implies $m_1(z^\gamma, w^\gamma, \lambda) \geq \min\{z^\gamma, w^\gamma\} \geq q^\gamma$, i.e., $m_1(x^\gamma, y^\gamma, \lambda) \in X(q, \gamma)$. When $\gamma < 0$, the map $u \in (0, 1] \mapsto u^\gamma$ is decreasing, so $u \geq q$ yields $u^\gamma \leq q^\gamma$. This in turn implies $m_1(z^\gamma, w^\gamma, \lambda) \leq \max\{z^\gamma, w^\gamma\} \leq q^\gamma$, i.e., $m_1(x^\gamma, y^\gamma, \lambda) \in X(q, \gamma)$. \square

3.4.3 Eventual Convexity of the Feasible Set

As in the result by Kataoka ([121]), convexity of the feasible set $M(p)$ cannot always be obtained for any probability level p . From a practical perspective this is not necessarily a problem since we are naturally looking for high p levels in problems of type (3.4.3). In some cases, we can show that the feasible set is convex if p is large enough. Convexity of feasible sets with high enough p is known as eventual convexity. In this section we will show that we can derive such eventual convexity if the Copulae, individual probability distribution functions and constraint mappings have some specific generalized concavity properties.

In the following theorem, we will provide conditions on Copulae, individual probability distribution functions and constraint mappings h such that eventual convexity of the feasible set can be asserted.

Theorem 3.4.18. *Let $\xi \in \mathbb{R}^m$ be a random vector with associated Copula C , and let $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ be functions such that*

$$\mathbb{P}[\xi \leq h(x)] = C(F_1(h_1(x)), \dots, F_m(h_m(x))), \quad (3.4.8)$$

where F_i are the marginal distribution functions of component i of ξ , $i = 1, \dots, m$. Assume that we can find $\alpha_i \in \mathbb{R}$, such that the functions h_i are α_i -concave and a second set of parameters $\gamma_i \in (-\infty, \infty]$, $b_i > 0$ such that either one of the following conditions holds:

1. $\alpha_i < 0$ and $z \mapsto F_i(z^{\frac{1}{\alpha_i}})$ is γ_i -concave on $(0, b_i^{\alpha_i}]$
2. $\alpha_i = 0$ and $z \mapsto F_i(\exp z)$ is γ_i -concave on $[\log b_i, \infty)$
3. $\alpha_i > 0$ and $z \mapsto F_i(z^{\frac{1}{\alpha_i}})$ is γ_i -concave on $[b_i^{\alpha_i}, \infty)$,

where $i \in \{1, \dots, m\}$ is arbitrary. If the Copula is either δ - γ -concave or δ - γ - $F(b)$ -concave for $\gamma \leq \gamma_i \leq \infty$, $i = 1, \dots, m$, then the set $M(p) := \{x \in \mathbb{R}^n : \mathbb{P}[\xi \leq h(x)] \geq p\}$ is convex for all $p > p^* := \max_{i=1, \dots, m} F_i(b_i)$. Convexity can moreover be derived for all $p \geq p^*$ if each individual distribution function F_i , $i = 1, \dots, m$ is strictly increasing. In the specific case that $\alpha_i \geq 0$, γ_i -concavity of the distribution functions holding everywhere, for all $i \in \{1, \dots, m\}$ and C being a δ - γ -concave Copula, the set $M(p)$ is convex for all p .

Proof. Pick any $p > p^*$, $x, y \in M(p)$, $\lambda \in [0, 1]$ and $i \in \{1, \dots, m\}$ arbitrarily. Define $x^\lambda := m_1(x, y, \lambda)$. Since all Copulae are dominated by the maximum-Copula, we get:

$$F_i(h_i(x)) \geq \min_{j=1, \dots, m} F_j(h_j(x)) \geq C(F_1(h_1(x)), \dots, F_m(h_m(x))) \geq p > p^* \geq F_i(b_i). \quad (3.4.9)$$

Now the latter entails

$$h_i(x) \geq b_i. \quad (3.4.10)$$

Estimate (3.4.10) also holds whenever $p \geq p^*$ and F_i is strictly increasing for each $i = 1, \dots, m$. A similar estimate is obtained for y clearly. We make a case distinction

1. $\alpha_i < 0$: In this case $\lambda h_i(x)^{\alpha_i} + (1 - \lambda)h_i(y)^{\alpha_i} \leq \max \{h_i(x)^{\alpha_i}, h_i(y)^{\alpha_i}\} \leq b_i^{\alpha_i}$
2. $\alpha_i = 0$: In this case $\lambda \log h_i(x) + (1 - \lambda) \log h_i(y) \geq \min \{\log h_i(x), \log h_i(y)\} \geq \log b_i$.
3. $\alpha_i > 0$: In this case $\lambda h_i(x)^{\alpha_i} + (1 - \lambda)h_i(y)^{\alpha_i} \geq \min \{h_i(x)^{\alpha_i}, h_i(y)^{\alpha_i}\} \geq b_i^{\alpha_i}$.

From monotonicity of the probability distribution function F_i , and α_i -concavity of g_i we obtain

$$F_i(h_i(x^\lambda)) \geq F_i(m_{\alpha_i}(h_i(x), h_i(y), \lambda)) = F_i((\lambda h_i(x)^{\alpha_i} + (1 - \lambda)h_i(y)^{\alpha_i})^{\frac{1}{\alpha_i}}), \quad (3.4.11)$$

whenever $\alpha_i \neq 0$ and

$$F_i(h_i(x^\lambda)) \geq F_i(m_0(h_i(x), h_i(y), \lambda)) = F_i(\exp(\lambda \log h_i(x) + (1 - \lambda) \log h_i(y))), \quad (3.4.12)$$

when $\alpha_i = 0$. The mappings in the right-hand side are γ_i concave by assumption on a specific domain. Since we have shown that our arguments map in this domain, we can apply γ_i -concavity and obtain:

$$F_i(h_i(x^\lambda)) \geq m_{\gamma_i}(F_i(h_i(x)), F_i(h_i(y)), \lambda). \quad (3.4.13)$$

Since i was fixed but arbitrary, the above equation holds for all $i = 1, \dots, m$.

A Copula is strictly increasing in its arguments, so we get from (3.4.13):

$$\begin{aligned} C(F_1(h_1(x^\lambda)), \dots, F_m(h_m(x^\lambda))) &\geq \\ C(m_{\gamma_1}(F_1(h_1(x)), F_1(h_1(y)), \lambda), \dots, m_{\gamma_m}(F_m(h_m(x)), F_m(h_m(y)), \lambda)) &\end{aligned} \quad (3.4.14)$$

If the Copula is δ - γ -concave everywhere then we can apply Lemma 3.4.16 to obtain:

$$\begin{aligned} C(m_{\gamma_1}(F_1(h_1(x)), F_1(h_1(y)), \lambda), \dots, m_{\gamma_m}(F_m(h_m(x)), F_m(h_m(y)), \lambda)) &\geq \\ m_\delta(C(F_1(h_1(x)), \dots, F_m(h_m(x))), C(F_1(h_1(y)), \dots, F_m(h_m(y))), \lambda) &\geq p, \end{aligned}$$

which together with (3.4.14) gives $C(F_1(h_1(x^\lambda)), \dots, F_m(h_m(x^\lambda))) \geq p$, i.e., $x^\lambda \in M(p)$. If the Copula is δ - γ - $F(b)$ -concave, we have $F_i(h_i(x)) \geq F_i(b_i)$ and we can apply Lemma 3.4.17 to obtain the same expression. Altogether we have shown:

$$\begin{aligned} C(F_1(h_1(x^\lambda)), \dots, F_m(h_m(x^\lambda))) &\geq \\ m_\delta(C(F_1(h_1(x)), \dots, F_m(h_m(x))), C(F_1(h_1(y)), \dots, F_m(h_m(y))), \lambda) &\geq p, \end{aligned}$$

which is equivalent with $C(F_1(h_1(x^\lambda)), \dots, F_m(h_m(x^\lambda))) \geq p$, i.e., $x^\lambda \in M(p)$ as was to be shown.

The asserted convexity of $M(p)$ regardless of p under the additional assumptions regarding the generalized concavity of h_i and F_i (for each $i = 1, \dots, m$) follows since one can apply the estimates (3.4.11) and (3.4.12) for all x, y , regardless of the domains to which $h_i(x)$ and $h_i(y)$ belong. All other estimates of this proof also carry through. Therefore, since the request $p \geq p^*$ was only needed to obtain (3.4.10), convexity of $M(p)$ can be shown regardless of p in that case. \square

An unfortunate effect in this lemma is that p^* depends somehow on the "worst" distribution function F_i and it is only needed to obtain estimate (3.4.9). We can actually sharpen the bound on p^* as the following theorem shows:

Theorem 3.4.19. *Define the set $D := \{x \in \mathbb{R}^n : h_i(x) \geq b_i, \forall i = 1, \dots, m\}$, where b_i is as defined in Theorem 3.4.18 and we make the same assumptions on ξ , F_i and the Copula. Then the set D is convex and $D \cap M(p)$ is convex for all $p \geq p^* = C(F_1(b_1), \dots, F_n(b_n))$.*

Proof. Since the mappings h_i are α_i -concave for all $i = 1, \dots, m$, the set D is indeed convex. Now from monotonicity of the Copula and the distribution functions F_i , $x \in D$ implies $C(F_1(h_1(x)), \dots, F_m(h_m(x))) \geq p^*$. Since we dispose of estimate (3.4.10), for any $x, y \in D$, the rest of the proof of Theorem 3.4.18 carries through. \square

Finally, we provide one last extension of the theorem with asymptotic δ -concave Copulae.

Theorem 3.4.20. *Let $\xi \in \mathbb{R}^m$ be a random vector with associated Copula C , and let $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ be functions such that*

$$\mathbb{P}[\xi \leq g(x)] = C(F_1(h_1(x)), \dots, F_m(h_m(x))), \quad (3.4.15)$$

where F_i are the marginal distribution functions of component i of ξ with unbounded support. Assume that we can find $\alpha_i \in \mathbb{R}$, such that the functions h_i are α_i -concave and a second set of parameters $\gamma_i \in [\bar{\gamma}, \infty]$, $b_i > 0$, $\bar{\gamma} > 0$ such that either one of the following conditions holds:

1. $\alpha_i < 0$ and $z \mapsto F_i(z^{\frac{1}{\alpha_i}})$ is γ_i -concave on $(0, b_i^{\alpha_i}]$
2. $\alpha_i = 0$ and $z \mapsto F_i(\exp z)$ is γ_i -concave on $[\log b_i, \infty)$
3. $\alpha_i > 0$ and $z \mapsto F_i(z^{\frac{1}{\alpha_i}})$ is γ_i -concave on $[b_i^{\alpha_i}, \infty)$,

for all $i \in \{1, \dots, m\}$. If the Copula is asymptotically δ -concave, then set $M(p) := \{x \in \mathbb{R}^n : \mathbb{P}[\xi \leq h(x)] \geq p\}$ is convex for all $p > p^* := \max_{i=1, \dots, m} F_i(b_i)$.

Proof. Pick $x, y \in M(p)$, $\lambda \in [0, 1]$ and $i \in \{1, \dots, m\}$ arbitrarily. Since the distribution functions F_i are assumed to have unbounded support, it follows that $F_i(h_i(x)) < 1$, $F_i(h_i(y)) < 1$ and $F_i(h_i(\lambda x + (1 - \lambda)y)) < 1$. From the definition of asymptotic δ -concavity, it follows that one can find a $\tilde{\gamma} > 0$ such that $F_i(h_i(z)) \leq q_i(\gamma)$ for all $\gamma < \tilde{\gamma}$, $z = x, y$ or $z = \lambda x + (1 - \lambda)y$ and for all $i = 1, \dots, m$. One can pick $\tilde{\gamma}$ moreover such that $\tilde{\gamma} \leq \bar{\gamma}$. This shows that all arguments in the proof of Theorem 3.4.18 are in the set $X(q, \gamma)$ for $\gamma < \tilde{\gamma}$ and one can therefore apply that theorem to conclude the proof. \square

Remark 3.4.21. *Our main Theorem 3.4.18 provides a link between generalized concavity requirements on the constraint mapping g , that of the one dimensional distribution functions F_i , $i = 1, \dots, m$ and that of the Copula. This link was less apparent in the earlier results [111] since it was required to have $\alpha_i < 0$, $\gamma_i = 1$ together with $\delta = \gamma = 0$. The results presented here show that already in the setting of [111] an improvement could be*

obtained by remarking that concavity of $z \mapsto F(z^{\frac{1}{\alpha}})$ for one dimensional probability distribution functions F on a set K implies log-concavity on the same set. The result is clearly an extension of those obtained in [110], since the independent Copula is 0-0-concave.

Remark 3.4.22. *The advantage of Theorem 3.4.19 over Theorem 3.4.18 is that it reinserts the dependence of p^* on the Copula and not just its generalized concavity property. We will illustrate this effect in later examples. If we call p_1^* the critical level obtained in Theorem 3.4.18 and p_2^* that of Theorem 3.4.19, then it follows always $p_2^* \leq p_1^*$. Indeed picking $p \geq p_1^*$, $x \in M(p)$ one derives from equation (3.4.10) that $x \in D$, i.e., $x \in M(p) \cap D$. The latter set is shown to be convex whenever $p \geq p_2^*$, implying $p_1^* \geq p_2^*$.*

An extension of interest is obtained only if either

1. More constraint mappings h can be allowed for. This is not truly the case, since any α -concave mapping is in particular β -concave for some $\beta < 0$. However we might be able to exploit better the true concavity properties of the mappings. This is shown in section 3.4.4.1.
2. The obtained probability level p^* is lower. This is shown in the examples of section 3.4.4.2.
3. The class of δ - γ -concave Copulae is larger than that of the log-exp-concave Copulae. This is shown in section 3.4.4.3.

These points are illustrated in section 3.4.4 through examples and results.

One key question if probability distribution functions with the required properties of Theorem 3.4.18 exist is answered positively already in the papers [110, 111]. It is shown that a specific property of the density known as r -decreasingness is sufficient (see Definition 2.5.21). Indeed this property induces, through Lemma 2.5.22 concavity of $z \mapsto F(z^{-\frac{1}{r}})$ on a set $(0, t^*)^{-r}$ for some $t^* > 0$.

It turns out that many distribution functions have this property and the dependence of t^* on r can be analytically computed for these distributions. We refer to table 1 of [111] for those results.

3.4.4 A generalization of Results

3.4.4.1 Improved use of Generalized Concavity of the Constraint Mappings

We begin by showing that Theorem 3.4.18 allows us to exploit the true generalized concavity properties of the constraint mappings h better. To this end consider the following example:

Example 3.4.23. *Consider log-concave mappings h_i and a log-exp-concave Copula together with a random vector having exponentially distributed components. Let λ denote the parameter of these exponential distributions. We can for instance take $h_1(x, y) =$*

$\log(x + y) + 0.1$ and $h_2(x, y) = \log(2x + 3y) + 0.2$, where $h_i : \mathbb{R}_+^2 \rightarrow \mathbb{R}$, $i = 1, 2$. In order to be able to apply Theorem 3.4.18 and derive convexity of the set $M(p)$ for chance constraints structured as (3.4.2), we have to show that the mappings $z \mapsto F(e^z)$ are log-concave, i.e., $z \mapsto f(z) := \log(1 - \exp(-\lambda e^z))$ has to be concave. This can be shown to hold on \mathbb{R} . Pick $x, y \in \mathbb{R}$ and $\mu \in [0, 1]$ arbitrarily. From convexity of the exponential function we get $-\lambda \exp(m_1(x, y, \mu)) \geq -\lambda m_1(\exp(x), \exp(y), \mu)$. A second application then yields

$$-\exp(-\lambda m_1(\exp(x), \exp(y), \mu)) \geq -m_1(\exp(-\lambda \exp(x)), \exp(-\lambda \exp(y)), \mu).$$

Altogether we get from strict increasingness of the log

$$\begin{aligned} \log(1 + (-\exp(-\lambda \exp(m_1(x, y, \mu)))))) &\geq \\ \log(m_1(1 - \exp(-\lambda \exp(x)), 1 - \exp(-\lambda \exp(y)), \mu)) &\geq m_1(f(x), f(y), \mu), \end{aligned}$$

as was required. We can therefore apply Theorem 3.4.18 to show that convexity of $M(p)$ holds for all p levels. Since any log-concave mapping is in particular r -concave for all $r < 0$, we could have also applied the earlier results obtained by [111]. Then we would obtain convexity of the feasible set for $p \geq 1 - e^{r-1}$. And this would hold for all $r < 0$, yielding convexity for $p > 1 - e^{-1} = 0.63$.

3.4.4.2 Improved Estimates of p^*

Following Remark 3.4.21, Theorem 3.4.18 appears in a somewhat weaker form in [110, 111]. In particular, in those papers the authors show that if a density is $r + 1$ -decreasing, then the constraint mapping $z \mapsto F(z^{-\frac{1}{r}})$ is concave on some set $(0, t_r^*)$. Now this corresponds to picking $\gamma_i = 1$ in Theorem 3.4.18, but log-exp concavity of Copulae implies $\gamma = 0$, leaving room for a gap to be filled. Indeed the following Lemma is a trivial consequence of generalized concavity:

Lemma 3.4.24. *Let $F : \mathbb{R} \rightarrow [0, 1]$ be a probability distribution function with a $\gamma + 1$ -decreasing density for some $\gamma > 0$. Then the function $z \mapsto F(z^{-\frac{1}{\gamma}})$ is α -concave on $(0, (t^\#)^{-\gamma})$, with $t^\# \leq t^*$, where t^* is as in Definition 2.5.21. Moreover this holds for all $\alpha \leq 1$.*

Proof. This follows trivially since α -concavity implies β -concavity for all $\beta \leq \alpha$ on the same set, so one can pick $t^\# \leq t^*$, potentially degenerate $t^\# = t^*$. \square

An example shows that weaker concavity is actually obtained on a larger set, i.e., $t^\# < t^*$.

Example 3.4.25. *We can get back to [110, example 4.1]. To this end we pick -1 -concave mappings h_i . In the above cited example the specific mappings $h_1(x, y) = \frac{1}{x^2 + y^2 + 0.1}$ and $h_2(x, y) = \frac{1}{(x+y)^2 + 0.1}$ from \mathbb{R}^2 to \mathbb{R} are chosen. We moreover pick any 0 - 0 -concave Copula, i.e., some log-exp concave Copula, such as for instance the independent, Gumbel or maximum Copula. In order to be able to apply Theorem 3.4.18 and derive convexity of the set $M(p)$ for chance constraints structured as (3.4.2), we have to show that*

$z \mapsto F(1/z)$ is log-concave on some set $(0, (t^\#)^{-1})$. Assume that F is the distribution function of an exponential random variable with parameter λ . Then upon defining the mapping $f(z) = \log F(1/z) = \log(1 - \exp(-\frac{\lambda}{z}))$, we have to show that this mapping is concave. To this end we will compute the first and second derivative and we obtain for any $z > 0$:

$$\begin{aligned} f'(z) &= (1 - (1 - \exp(-\frac{\lambda}{z}))^{-1})\lambda z^{-2} \\ f''(z) &= f'(z)[-2z^{-1} + \lambda z^{-2}(1 - \exp(-\frac{\lambda}{z}))^{-1}]. \end{aligned}$$

Now $z > 0$ implies $0 < \exp(-\frac{\lambda}{z}) < 1$, yielding $(1 - \exp(-\frac{\lambda}{z}))^{-1} > 1$ and therefore $f'(z) < 0$. So the sign of $f''(z)$ depends on that of $[-2z^{-1} + \lambda z^{-2}(1 - \exp(-\frac{\lambda}{z}))^{-1}]$. From the above estimate we get

$$-2z^{-1} + \lambda z^{-2}(1 - \exp(-\frac{\lambda}{z}))^{-1} \geq -2z^{-1} + \lambda z^{-2}, \quad (3.4.16)$$

so indeed $f''(z) < 0$ for small z (but we knew this already, since the exponential density is 2-decreasing). It turns out that $f''(z) = 0$ if and only if $\exp(-\frac{\lambda}{z}) = 1 - \frac{\lambda}{2z}$. We can compute this $z^\#$ by using for instance a dichotomy procedure (picking $\lambda = 1$, yielding $z^\# = 0.62750048$, i.e., $t^\# = 1.59362426$). The obtained p^* of Theorem 3.4.18 is then equal to $p^* = 1 - \exp(-\lambda t^\#) = 0.7968121$, which is significantly better than the earlier obtained $p^* = 0.864$. Empirically varying λ yields the result that the improved p^* does not depend on λ , similarly to the results obtained in [110].

We can also look at Theorem 3.4.19 and look at the set $M(p) \cap \{x \in \mathbb{R}^n : h_i(x) \geq t^\#\}$ for $p \geq p^* = C(F_1(t^\#), \dots, F_m(t^\#))$. Now this results depends on the Copula. Picking the Maximum Copula we get $p^* = 0.7968121$ as before. Picking the Independent Copula we get $p^* = 0.7968121^2 = 0.6349$ and the Gumbel Copula with $\theta = 1.1$ yields $p^* = 0.652770$. The latter result does not depend on θ , at least as found empirically.

Example 3.4.26. Returning once again to [110, example 4.1] and example 3.4.25 above, we can likewise stipulate that the components of ξ follow a standard normal distribution. These components are linked through a log-exp concave Copula. In the paper [110] the authors have shown that the map $z \mapsto \Phi(1/z)$ is concave on $(0, 1/\sqrt{2})$. This then yields a bound of $p^* = \Phi(\frac{1}{\sqrt{2}}) = 0.921$, where Φ is the probability distribution function of a standard normal random variable in dimension 1. In order to be able to apply Theorem 3.4.18 and derive convexity of the set $M(p)$ for chance constraints structured as (3.4.2), we have to show that $z \mapsto \Phi(1/z)$ is log-concave only. Calling $f(z) = \log(\Phi(1/z))$, we can compute and obtain for $z > 0$:

$$\begin{aligned} f'(z) &= -z^{-2}\Phi'(z^{-1})\Phi(z^{-1})^{-1} \\ f''(z) &= f'(z^{-1})(-2z^{-1} + z^{-3} + z^{-2}\Phi'(z^{-1})\Phi(z^{-1})^{-1}), \end{aligned}$$

where we have used that $\Phi''(z^{-1}) = -z^{-1}\Phi'(z^{-1})$. Indeed, $f'(z) < 0$ for $z > 0$ and $f''(z) < 0$ for $z > 0$ small enough. We can again numerically compute $z^\#$ such that $f''(z^\#) = 0$, i.e., $-2z^{-1} + z^{-3} + z^{-2}\Phi'(z^{-1})\Phi(z^{-1})^{-1} = 0$ at $z = z^\#$, and obtain $z^\# =$

0.754205. Convexity of the feasible set is then obtained for $p \geq p^* = \Phi(\frac{1}{z^\#}) = 0.90756$, from Theorem 3.4.18 which is only marginally better.

Applying Theorem 3.4.19 however and distinguishing Copulae allows us to obtain an improved p^* level for the Gumbel and Independent Copula. Indeed, then we get $p^* = 0.8334$ and $p^* = 0.824$ respectively.

3.4.4.3 More Copulae

The last extension that Theorem 3.4.18 allows for is that we can use more Copulae, if the class of δ - γ -concave Copulae contains more Copulae than just log-exp-concave Copulae. This turns out to be the case. Indeed the following Lemma shows that the Clayton Copula is δ - γ -concave for specific δ values. It was however shown in [111] that it was not log-exp-concave.

Lemma 3.4.27. *Let $\theta > 0$ be the parameter of the strict generator $\psi : [0, 1] \rightarrow \mathbb{R}_+$, $\psi(t) = \theta^{-1}(t^{-\theta} - 1)$ of the Clayton Copula. This Copula is δ - γ -concave for all $\gamma > 0$ provided that $\delta \leq -\theta < 0$.*

Proof. The inverse of the generator is given by $\psi^{-1}(s) = (\theta s + 1)^{-\frac{1}{\theta}}$, so the Copula is defined as $C(u) = \psi^{-1}(\sum_{i=1}^n \psi(u_i))$, where $u \in [0, 1]^m$. We begin by computing the derivatives of the generator

$$\begin{aligned} \frac{d\psi}{dt}(t) &= -t^{-\theta-1} \\ \frac{d\psi^{-1}}{ds}(s) &= (\theta s + 1)^{-\frac{1}{\theta}-1} = -\psi^{-1}(s)(\theta s + 1)^{-1}. \end{aligned}$$

We will consider the mapping $f(u) = C(u^{\frac{1}{\gamma}})^\delta$ and have to show that it is convex. From Remark 3.4.3 it follows that it is sufficient to consider $u \in (0, 1]^m$ only. This implies that $C(u^{\frac{1}{\gamma}}) > 0$ and in particular that $f(u)$ and its derivatives are well defined. We therefore fix $u \in (0, 1]^m$ arbitrarily and compute the first and second order derivatives of f . A computation gives

$$\frac{\partial f}{\partial u_k}(u) = \delta C(u^{\frac{1}{\gamma}})^{\delta-1} \frac{\partial C}{\partial z_k}(u^{\frac{1}{\gamma}}) \frac{1}{\gamma} u_k^{\frac{1}{\gamma}-1}, k = 1, \dots, m. \quad (3.4.17)$$

When computing the derivative of the Copula, we obtain:

$$\frac{\partial C}{\partial u_k}(u) = \frac{d\psi^{-1}}{ds} \left(\sum_{i=1}^m \psi(u_i) \right) \frac{d\psi}{dt}(u_k) = C(u) (\theta \sum_{i=1}^m \psi(u_i) + 1)^{-1} u_k^{-\theta-1}, k = 1, \dots, m \quad (3.4.18)$$

Substituting altogether we get:

$$\frac{\partial f}{\partial u_k}(u) = \frac{\delta}{\gamma} C(u^{\frac{1}{\gamma}})^\delta (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-1} u_k^{-\frac{\theta+\gamma}{\gamma}}, k = 1, \dots, m. \quad (3.4.19)$$

Deriving a second time, we get

$$\begin{aligned}
\frac{\partial^2 f}{\partial u_k^2}(u) &= \frac{\delta}{\gamma}(\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-1} u_k^{-\frac{\theta+\gamma}{\gamma}} \frac{\partial}{\partial u_k} C(u^{\frac{1}{\gamma}})^\delta \\
&+ \frac{\delta}{\gamma} C(u^{\frac{1}{\gamma}})^\delta u_k^{-\frac{\theta+\gamma}{\gamma}} \frac{\partial}{\partial u_k} (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-1} \\
&+ \frac{\delta}{\gamma} C(u^{\frac{1}{\gamma}})^\delta (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-1} \frac{\partial}{\partial u_k} u_k^{-\frac{\theta+\gamma}{\gamma}}, k = 1, \dots, m.
\end{aligned}$$

In the first line we can just substitute (3.4.19), in the second line we obtain

$$\begin{aligned}
\frac{\partial}{\partial u_k} (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-1} &= -(\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-2} \theta \frac{d\psi}{dt}(u_k^{\frac{1}{\gamma}}) \frac{1}{\gamma} u_k^{\frac{1}{\gamma}-1} \\
&= \frac{\theta}{\gamma} (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-2} u_k^{-\frac{\theta+\gamma}{\gamma}}, k = 1, \dots, m.
\end{aligned}$$

and in the third line,

$$\frac{\partial}{\partial u_k} u_k^{-\frac{\theta+\gamma}{\gamma}} = -\frac{\theta + \gamma}{\gamma} u_k^{-\frac{\theta+\gamma}{\gamma}-1}, k = 1, \dots, m. \quad (3.4.20)$$

For the cross-derivative, $j, k = 1, \dots, m$, the expression becomes:

$$\begin{aligned}
\frac{\partial^2 f}{\partial u_j \partial u_k}(u) &= \frac{\delta}{\gamma} (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-1} u_k^{-\frac{\theta+\gamma}{\gamma}} \frac{\partial}{\partial u_j} C(u^{\frac{1}{\gamma}})^\delta \\
&+ \frac{\delta}{\gamma} C(u^{\frac{1}{\gamma}})^\delta u_k^{-\frac{\theta+\gamma}{\gamma}} \frac{\partial}{\partial u_j} (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-1}, j \neq k
\end{aligned}$$

In the first line one substitutes (3.4.19) with k replaced by j and the second is dealt with similarly as before. Combining these expression we obtain for all $u \in (0, 1]^m$, $k, j = 1, \dots, m$:

$$\begin{aligned}
\frac{\partial^2 f}{\partial u_k^2}(u) &= \frac{\delta^2 + \delta\theta}{\gamma^2} C(u^{\frac{1}{\gamma}})^\delta (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-2} (u_k^{-\frac{\theta+\gamma}{\gamma}})^2 \\
&- \frac{\delta}{\gamma} \frac{\theta + \gamma}{\gamma} C(u^{\frac{1}{\gamma}})^\delta (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-1} u_k^{-\frac{\theta+\gamma}{\gamma}-1} \\
\frac{\partial^2 f}{\partial u_j \partial u_k}(u) &= \frac{\delta^2 + \delta\theta}{\gamma^2} C(u^{\frac{1}{\gamma}})^\delta (\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1)^{-2} u_k^{-\frac{\theta+\gamma}{\gamma}} u_j^{-\frac{\theta+\gamma}{\gamma}}, j \neq k.
\end{aligned}$$

We have to show that the Hessian is positive semi-definite for all $u \in (0, 1]^m$, we will do this directly, by picking $z \neq 0, z \in \mathbb{R}^m$ and forming $z^T \nabla^2 f(u) z$. The special structure

of the above second-derivatives is as follows $\frac{\partial^2 f}{\partial u_k^2}(u) = \alpha(u)x_k^2 + \beta_k(u)$ and $\frac{\partial^2 f}{\partial u_j \partial u_k}(u) = \alpha(u)x_j x_k$. Now

$$\begin{aligned}
z^\top \nabla^2 f(u) z &= \sum_{j=1}^m \sum_{k=1}^m z_j \frac{\partial^2 f}{\partial u_j \partial u_k}(u) z_k \\
&= \sum_{j,k,j \neq k} \alpha(u) z_j x_j x_k z_k + \sum_{k=1}^m \alpha(u) z_k^2 x_k^2 + z_k^2 \beta_k(u) \\
&= \alpha(u) \sum_{j=1}^m \sum_{k=1}^m z_j x_j x_k z_k + \sum_{k=1}^m z_k^2 \beta_k(u) \\
&= \alpha(u) \left(\sum_{j=1}^m z_j x_j \right)^2 + \sum_{k=1}^m z_k^2 \beta_k(u).
\end{aligned}$$

Substituting the above expressions, we obtain:

$$\begin{aligned}
z^\top \nabla^2 f(u) z &= C(u^{\frac{1}{\gamma}})^\delta \frac{\delta^2 + \delta\theta}{\gamma^2} \left(\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1 \right)^{-2} \left(\sum_{i=1}^m z_i u_i^{-\frac{\theta+\gamma}{\gamma}} \right)^2 \\
&\quad - C(u^{\frac{1}{\gamma}})^\delta \frac{\delta}{\gamma} \frac{\theta + \gamma}{\gamma} \left(\theta \sum_{i=1}^m \psi(u_i^{\frac{1}{\gamma}}) + 1 \right)^{-1} \sum_{i=1}^m z_i^2 u_i^{-\frac{\theta+\gamma}{\gamma}-1}. \quad (3.4.21)
\end{aligned}$$

Now with our choice of δ , we obtain $\delta^2 + \delta\theta \geq 0$ and similarly $-\frac{\delta}{\gamma} \geq 0$. Together with $u \in (0, 1]^m$, one can see that all expressions in (3.4.21) are positive. This therefore yields $z^\top \nabla^2 f(u) z \geq 0$, i.e., f is a convex function, as was to be shown. \square

Remark 3.4.28. *Numeric evidence would indicate that the Clayton Copula is also δ - γ -concave for δ slightly bigger than $-\theta$, but not for all $\delta < 0$. Indeed, in dimension 2, picking $\delta = -0.03$, $\theta = 0.1$, $\gamma = 0.5$ and evaluating the above Hessian at the point $u = (0.96, 0.985)$, one obtains a negative eigenvalue.*

Remark 3.4.29. *We can also prove that the Clayton Copula is δ -0 concave for $\delta \leq -\theta$. Indeed setting $f(u) = C(e^u)^\delta$ one obtains for $u \in (-\infty, 0]^m$:*

$$\frac{\partial f}{\partial u_k}(u) = \delta C(e^u)^\delta \left(\theta \sum_{i=1}^m \psi(e^{u_i}) + 1 \right)^{-1} e^{-\theta u_k}, \quad k = 1, \dots, m$$

and for $j, k = 1, \dots, m$:

$$\begin{aligned}
\frac{\partial^2 f}{\partial u_k^2}(u) &= (\delta^2 + \delta\theta) C(e^u)^\delta \left(\theta \sum_{i=1}^m \psi(e^{u_i}) + 1 \right)^{-2} e^{-2\theta u_k} \\
&\quad - \theta \delta C(e^u)^\delta \left(\theta \sum_{i=1}^m \psi(e^{u_i}) + 1 \right)^{-1} e^{-\theta u_k} \\
\frac{\partial^2 f}{\partial u_j \partial u_k}(u) &= (\delta^2 + \delta\theta) C(e^u)^\delta \left(\theta \sum_{i=1}^m \psi(e^{u_i}) + 1 \right)^{-2} e^{-\theta u_k} e^{-\theta u_j}, \quad j \neq k.
\end{aligned}$$

And the same results follow.

We have shown in lemma 3.4.27 that the Clayton Copula is δ - γ -concave for all $\gamma > 0$ and $\delta \leq -\theta$. Since this Copula is not log-exp concave, the results of [111] could not be applied. We can however use our theorem to derive convexity of feasible sets $M(p)$. As such we get the example:

Example 3.4.30. *Consider again the same setting as that of example 3.4.23, except that this time we use the Clayton Copula to link the components of ξ together. Since this Copula is δ - γ -concave for any $\gamma > 0$ and $\delta \leq -\theta$, we have to show that the mappings $z \mapsto F(1/z)$ are γ -concave on some set $(0, (t^\#)^{-1})$. Since results hold in particular for $\gamma < 1$, concavity of those maps suffices. That is obtained whenever their densities are for instance 2-decreasing. Assuming that ξ follows an exponential distribution with parameter λ , we obtain the very rough bound $p^* = 1 - e^{-2} = 0.864$. But we can do better, to this end we have to show that the mapping $f(z) = F(1/z)^\gamma = (1 - \exp(-\frac{\lambda}{z}))^\gamma$, is concave. A computation gives*

$$\begin{aligned} f'(z) &= -\lambda\gamma z^{-2}(1 - \exp(-\lambda z^{-1}))^{\gamma-1} \exp(-\lambda z^{-1}) \\ f''(z) &= f'(z)[-2z^{-1} - (\gamma - 1)\lambda z^{-2}(1 - e^{-\lambda z^{-1}})^{-1}e^{-\lambda z^{-1}} + \lambda z^{-2}]. \end{aligned}$$

Applying Theorem 3.4.18, and picking for instance $\gamma = \frac{1}{2}$, one obtains $z^\# = 0.54807$, giving the estimate $p^* = 0.8387$. With $\gamma = 0.01$, one obtains $z^\# = 0.62537$ and the estimate $p^* = 0.7979$. Again p^* does not depend on λ .

We can also apply Theorem 3.4.19 to obtain $p^* = 0.638$ for $\theta = 0.1$. This time the result depends on θ as can be shown empirically.

3.4.5 A Partial Characterization of the Gaussian Copula

So far we have only provided examples of Archimedean Copulae with the δ - γ -concavity property. In this section we investigate the Gaussian Copula and provide a partial characterization of the δ - γ -concavity properties of this Copula.

Lemma 3.4.31. *Let R be an $m \times m$ correlation matrix. Assume furthermore that the $m \times m$ inverse matrix is such that $Q := R^{-1} - I$ has only positive components. Let $\delta = \min_{i=1, \dots, m} \max_{j=1, \dots, m} Q_{ij}$ and define $q = \max \{ \log(\Phi(\frac{1}{\delta})), \log(\Phi(1)) \}$, then the density of the Gaussian Copula is 0-0-concave for all $u \in [q, 0]^m$, where Φ is the probability distribution function of a standard normal random variable.*

Proof. Define the mapping $f : (-\infty, 0]^m \rightarrow \mathbb{R}^m$ by setting

$$f(u) = \Phi^{-1}(e^u) = (\Phi^{-1}(e^{u_1}), \dots, \Phi^{-1}(e^{u_m})),$$

where Φ^{-1} is the inverse of the standard normal distribution function. We wish to show that the density of the Gaussian Copula wherein we substitute e^u , is eventually log-concave, i.e., the mapping $u \mapsto c_R(e^u)$ is concave. Now the density c_R of the Gaussian Copula is given by

$$c_R(u) = \frac{1}{\sqrt{\det R}} \exp(-0.5\Phi^{-1}(u)^\top(R^{-1} - I)\Phi^{-1}(u)). \quad (3.4.22)$$

This boils down to showing that the mapping $u \mapsto \hat{f}(u) := f(u)^\top (R^{-1} - I)f(u)$ is convex.

We will begin by computing some derivatives. To this end, fix any $u \in (-\infty, 0]^m$ completely arbitrarily. For convenience we will note $Q := R^{-1} - I$. We begin by computing the first and second derivative of a component of the vector $f(u)$, i.e., we compute the derivatives of the 1-dimensional mapping $v \in (-\infty, 0] \mapsto f_i(v)$ for arbitrary $i = 1, \dots, m$:

$$\frac{df_i}{dv}(v) = \frac{d\Phi^{-1}}{dv}(e^v)e^v = \sqrt{2\pi} \exp\left(\frac{1}{2}\Phi^{-1}(e^v)^2\right)e^v, \quad i = 1, \dots, m, \quad \forall v \in (-\infty, 0]$$

and

$$\begin{aligned} \frac{d^2\Phi^{-1}}{ds^2}(s) &= \sqrt{2\pi} \exp\left(\frac{1}{2}\Phi^{-1}(s)^2\right) \frac{1}{2} 2\Phi^{-1}(s) \frac{d\Phi^{-1}}{ds}(s) \\ &= (\sqrt{2\pi} \exp\left(\frac{1}{2}\Phi^{-1}(s)^2\right))^2 \Phi^{-1}(s), \quad \forall s \in (0, 1). \end{aligned}$$

Giving

$$\begin{aligned} \frac{d^2 f_i}{dv^2}(v) &= \frac{d^2\Phi^{-1}}{ds^2}(e^v)e^v e^v + \frac{df_i}{dv}(v) \\ &= f_i(v) \left(\frac{df_i}{dv}(v)\right)^2 + \frac{df_i}{dv}(v), \quad i = 1, \dots, m, \quad \forall v \in (-\infty, 0]. \end{aligned}$$

Now we can compute the derivatives of the mapping \hat{f} ,

$$\frac{\partial \hat{f}}{\partial u_k}(u) = 2 \frac{df_k}{du}(u_k) \sum_{j=1}^m Q_{kj} f_j(u_j), \quad k = 1, \dots, m$$

and the second derivatives are:

$$\begin{aligned} \frac{\partial^2 \hat{f}}{\partial u_k^2}(u) &= 2 \frac{d^2 f_k}{du^2}(u_k) Q_{kk} f_k(u_k) + 2 \frac{df_k}{du}(u_k) Q_{kk} \frac{df_k}{du}(u_k) \\ &\quad + 2 \sum_{j=1, j \neq k}^m \frac{d^2 f_k}{du^2}(u_k) Q_{kj} f_j(u_j), \quad k = 1, \dots, m \\ \frac{\partial^2 \hat{f}}{\partial u_j \partial u_k}(u) &= 2 \frac{df_k}{du}(u_k) Q_{kj} \frac{df_j}{du}(u_j), \quad j, k = 1, \dots, m, \quad j \neq k \end{aligned}$$

Substituting in the above expression the previously found identity for the second derivative of f_i , we obtain

$$\begin{aligned} \frac{\partial^2 \hat{f}}{\partial u_k^2}(u) &= 2 \frac{df_k}{du}(u_k) Q_{kk} \frac{df_k}{du}(u_k) + 2 \sum_{j=1}^m \frac{d^2 f_k}{du^2}(u_k) Q_{kj} f_j(u_j) \\ &= 2 \frac{df_k}{du}(u_k) Q_{kk} \frac{df_k}{du}(u_k) \\ &\quad + 2 \frac{df_k}{du}(u_k)^2 \left[(f_k(u_k) + \left(\frac{df_k}{du}(u_k)\right)^{-1}) \sum_{j=1}^m Q_{kj} f_j(u_j) \right], \quad k = 1, \dots, m \end{aligned}$$

where we have used that $\frac{df_k}{du}(u_k) > 0$ for all $u \in (-\infty, 0]^m$. Now picking $z \in \mathbb{R}^m$ and forming $z^\top \nabla^2 h(u) z$, one obtains the following

$$\begin{aligned} z^\top \nabla^2 \hat{f}(u) z &= 2 \sum_{i,j=1}^m z_i \frac{df_i}{du}(u_i) Q_{ij} \frac{df_j}{du}(u_j) z_j \\ &+ 2 \sum_{k=1}^m z_k^2 \frac{df_k}{du}(u_k)^2 \left[(f_k(u_k) + (\frac{df_k}{du}(u_k))^{-1}) \sum_{j=1}^m Q_{kj} f_j(u_j) \right], \end{aligned}$$

which holds for all $u \in (-\infty, 0]^m$ as u was chosen arbitrarily.

If we define the vector $x(u) \in \mathbb{R}^m$ as $x(u) = (z_1 \frac{df_1}{du}(u_1), \dots, z_m \frac{df_m}{du}(u_m))$, and the matrix $\tilde{Q}(u)$ as

$$\tilde{Q}(u)_{ij} = \begin{cases} Q_{ij} & \text{if } i, j = 1, \dots, m, i \neq j \\ Q_{ii} + (f_i(u_i) + (\frac{df_i}{du}(u_i))^{-1}) \sum_{j=1}^m Q_{ij} f_j(u_j) & \text{if } i = 1, \dots, m \end{cases}$$

It is clear that $z^\top \nabla^2 \hat{f}(u) z = x(u)^\top \tilde{Q}(u) x(u)$ and $z^\top \nabla^2 \hat{f}(u) z \geq 0$ if and only if $\tilde{Q}(u)$ is positive semi-definite. Defining the vector $\alpha(u) \in \mathbb{R}^m$ as the diagonal of $\tilde{Q}(u)$ minus the diagonal of Q , it is clear that $\tilde{Q}(u) = Q + \text{diag } \alpha(u)$, where $\text{diag } \alpha(u)$ is the diagonal matrix with elements of the vector $\alpha(u)$. It therefore follows that the eigenvalues of $\tilde{Q}(u)$ are those of R^{-1} to which we add $\alpha(u) - 1$. Making sure that $\alpha_i(u) - 1 \geq 0, \forall i = 1, \dots, m$ is therefore sufficient for $\tilde{Q}(u)$ to be positive semi-definite. From the conditions of the lemma it follows that $\sum_{j=1}^m Q_{ij} f_j(u_j) \geq \delta f_{j^*}(u_{j^*}), \forall i = 1, \dots, m$, where j^* is that element of i -th line of Q with $Q_{ij^*} \geq \delta$. Since the mapping f is strictly increasing, it follows $\sum_{j=1}^m Q_{ij} f_j(u_j) \geq \delta f_{j^*}(q) \geq 1, \forall i = 1, \dots, m$. The choice of q also implies $f_j(u_j) \geq 1, \forall j = 1, \dots, m$. Altogether we obtain $\alpha_i(u) \geq 1, i = 1, \dots, m$ for all $u \geq q$ and the result follows. \square

Remark 3.4.32. *The condition that $Q = R^{-1} - I$ needs to have only positive components is not very restrictive. For instance if $m = 2$, this implies that the off-diagonal element needs to be non-positive. When $m = 3$ such a simple characterization is not possible any longer. To show that cases exist satisfying the condition on Q ($m \geq 3$), we can consider as an example the three dimensional matrix with $R_{12} = -0.9, R_{13} = -0.7$ and $R_{23} = 0.5$.*

Example 3.4.33. *Let R be the correlation matrix in dimension \mathbb{R}^2 having -0.9 on the off-diagonal. Then $\delta = 4.74$ and $q = \max\{-0.54, -0.17\}$. In fact for all 2-dimensional matrices with off-diagonal element $\rho \leq \frac{1-\sqrt{5}}{2}$, this same bound holds.*

Lemma 3.4.34. *Let R be an m times m correlation matrix. Assume furthermore that the inverse matrix is such that $Q := R^{-1} - I$ has only positive components. Let $\delta = \min_{i=1, \dots, m} \min_{j=1, \dots, m} Q_{ij} > 0$ and define $q = \max\{\log(\Phi(\frac{1}{\delta})), \log(\Phi(1))\}$, where Φ is the probability distribution function of a standard normal random variable. Then the Gaussian Copula admits the following decomposition*

$$C(e^u) = C(e^q) + \sum_{i=2}^{2^m} \varphi_i(u),$$

where each function $\varphi_i : [0, 1]^m \rightarrow [0, 1]$ is a 0-concave probability distribution up to a multiplicative constant, $i = 2, \dots, 2^m$. This decomposition holds for all $u \in [q, 0]^m$.

Proof. In the proof of Lemma 3.4.31 we have seen that the density of the Gaussian Copula is "eventually" 0-0-concave. As a matter of fact, when looking at the final estimates, one can also derive $\sum_{j=1}^m Q_{ij} f_j(u_j) \geq \delta f_j(u_j)$, $\forall i = 1, \dots, m$, $\forall j = 1, \dots, m$, since this time δ is defined differently. This shows that this density is 0-0-concave, whenever $u_j \geq q$ for a single $j \in \{1, \dots, m\}$. Now let C_1, \dots, C_n , with $n = 2^m$ be the partition of $[0, 1]^m$ into orthants based at q , i.e., $C_1 = \{x \in (-\infty, 0]^m, x \leq q\}$, $C_n = \{x \in (-\infty, 0]^m, x \geq q\}$ etc... Then the density of the Gaussian Copula has been shown to be 0-0-concave on all sets C_2, \dots, C_n . Furthermore each C_i is a convex set. Hence the restriction of the density of $C(e^u)$ to each C_i , $i = 2, \dots, n$ is a log concave density by Corollary 3.4.15. Let \hat{f}_i denote the restriction of the density of $u \mapsto C(e^u)$ to C_i , $i = 1, \dots, n$. Defining $\varphi_i : [0, 1]^m \rightarrow [0, 1]$ as the integral of this restricted density \hat{f}_i , it follows by Theorem 4.15 [51] that φ_i is log-concave, $i = 2, \dots, n$. Now clearly

$$C(e^u) = \sum_{i=1}^n \int_{\prod_{j=1}^m (-\infty, u_j]} \hat{f}_i(s) ds = \int_{\prod_{j=1}^m (-\infty, u_j]} \hat{f}_1(s) ds + \sum_{i=2}^n \varphi_i(u).$$

If $u \in [q, 0]^m$, the first element is nothing else but $C(e^q)$, since $C(e^q) = \int_{\prod_{j=1}^m (-\infty, q_j]} \hat{f}_1(s) ds = \int_{C_1} \hat{f}_1(s) ds$. \square

Example 3.4.35. Let R be the correlation matrix in dimension \mathbb{R}^2 having -0.9 on the off-diagonal. Then $\delta = 4.263$ and $q = \max\{-0.526, -0.17\}$

3.4.6 A Potential Application

In some engineering problems in energy one easily stumbles across mixed laws in "columns". In particular, offer demand equilibrium constraints in unit commitment require that we commit a production schedule producing enough energy in most situations. However, uncertainty is only discovered later. This uncertainty consists of load uncertainty and uncertainty on renewable generation such as wind power. Wishing to produce enough electricity in most situations for all time steps simultaneously would then result in a constraint of the type

$$p \leq \mathbb{P}[\xi + \eta \leq h(x)], \tag{3.4.23}$$

where $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is the mapping associating with a decision vector $x \in \mathbb{R}^n$ its actual production level and $\xi \in \mathbb{R}^m$ and $-\eta \in \mathbb{R}^m$ are two random vectors, modeling for instance load uncertainty and wind generation respectively. Now to show that Copulae can be used to obtain a convex model requires an additional result. Theorem 4.2.3. of [181] (dating back to [41, 21, 22], [213, proof in dimension 1]) indicates that the convolution of log-concave densities is again a log-concave density. As such, picking each ξ_i, η_i individually following a log-concave density, it follows that their sum follows a log-concave density, i.e., $F_i(z) = \mathbb{P}[\xi_i + \eta_i \leq z]$ is a log-concave function. If we now use a δ - γ -concave Copula

with $\gamma \leq 0$ and concave functions $h_i(x)$, the set defined by (3.4.23) will then turn out to be eventually convex. Since in practice h_i is often linear, as a sum of production levels, it will be concave. Hence we can come up with a tractable model for such a mixed law setting.

Chapter 4

Algorithms for (convex) Probabilistic Programming

Optimization problems involving probabilistic constraints of type (2.0.1) can be transformed into convex optimization problems under some additional assumptions. For example if g is jointly quasi-concave and ξ has a log-concave density, then Theorem 2.5.15 allows us to assert that $x \mapsto \log(p) - \log(\mathbb{P}[g(x, \xi) \geq 0])$ is a convex mapping. Under such assumptions solving the probabilistically constrained optimization problem amounts to solving a convex constrained problem. However as indicated by J. Mayer ([149]) the probabilistic constraint, even if continuously differentiable, might be stiff. It is therefore preferable to consider the tools from non-smooth optimization. An elementary, but convincing example of this statement is the optimization of the stiff differentiable "maxanal" function ([20, Section 12.1.2]). It turns out that methods from smooth optimization are outperformed by methods from non-smooth analysis (e.g., bundle methods).

Returning to optimization problems with probabilistic constraints, classically only the supporting hyperplane method has been employed. Now the bottle-neck resides in the computation of the (sub-)gradient. This is not only time-consuming but can also only be done up to some precision. The latter precision can be controlled but a trade-off with the aforementioned computational effort has to be found. These gradients can be used to compute a linearization of the convex constraint induced by the probabilistic constraint. However, in general, we can not assert that the derived linearization remains below the convex mapping. The latter feature leads to the definition of a so-called upper oracle in section 4.1. A proximal and level bundle method for dealing with convex constrained optimization involving such upper-oracles are derived in sections 4.1 and 4.2. As a consequence a framework is provided for solving (convex) probabilistically constrained optimization problems efficiently.

4.1 A proximal Bundle Method

Real-life problems are often modeled in an uncertain setting, to better reflect unknown phenomena specific to the application. In particular, such is the case in the energy sector; see [210]. For the numerical experience of this section¹ we focus on a specific energy problem arising in stochastic unit commitment ([31], [225], [165], [255],[246]). This is the problem of optimally managing reservoirs of a hydro valley in the short term.

A hydro valley is a set of power plants cascaded along the same hydrological bassin. For the considered system, part of Electricité de France mix, uncertainty is mostly related to the amount of melted snow arriving as a streamflow to the most uphill reservoirs. The volume of these reservoirs changes with the inflows and determines the amount of water that can be converted into energy. After turbinning water to produce power, the upstream reservoirs release a certain volume that fills the reservoirs downstream, and the process continues until the power plant at the bottom of the bassin. In this interconnected context, it is important to jointly manage the generation of the cascaded plants in a manner that not only is economical but also reliable. More precisely, it is crucial to keep the volume of each reservoir in the valley between prescribed minimum and maximum levels (to prevent floods, to ensure touristic activities, etc). Since it is not realistic to ensure such conditions for every possible streamflow, satisfaction of lower and upper bounds for the volumes can be required in a probabilistic manner.

Introduced by [32], probability constraints are quite an appealing tool for dealing with uncertainty, because they give a physical interpretation to risk. For hydro valley management, chance constraints have been employed in [143, 63, 65, 142, 155, 265, 264, 244]. Most of these works require each component of the uncertain constraint to be satisfied in a probabilistic sense in a separate manner. As explained in [244], a stochastic model with such individual chance constraints may sometimes result in unreliable optimal decisions, because there is no guarantee that the whole stochastic inequality will be satisfied with a given probability. In this work we build upon the model in [244], with joint chance constraints, and derive a sound numerical solution procedure, based on bundle methods, [114, 20].

In the classical textbook [181], convexity of chance constraints is ensured for a variety of distributions. Accordingly, for appropriate choices of the stochastic model for the inflows (cf. Section 5.1 for details), the hydro valley management problem has the abstract form

$$\min_{x \in X \subseteq \mathbb{R}^n} \{f(x) : c(x) \leq 0\} \quad (4.1.1)$$

for f and c finite-valued convex continuous functions and X a compact convex polyhedron. Even when c can be shown to be differentiable ([244]), it is reported in [149] that joint chance constraints are occasionally sufficiently stiff for smooth optimization methods to encounter convergence issues. Moreover in many applications c is actually non-smooth ([106]). For this reason it is of interest to make no smoothness assumption on c . For algorithmical purposes, given any $x \in X$, joint chance constraints (corresponding to c in (4.1.1)) need to be efficiently computed, together with a gradient.

¹See also section 5.1

formulæ for multi-variate Gamma, Dirichlet, and Gaussian distributions can be found in [186], [181, 88, 221], and [181, 106, 244], respectively. We also refer to Section 2.7. Similarly to evaluating the function, these formulæ involve computing a probability. In turn, the calculation of a probability amounts to compute, for any given x , an integral in relatively high dimension (for our numerical application in Section 4.1.6, the corresponding dimension is 48). For multi-variate Gaussian distributions, the code developed by A. Genz can be used to efficiently approximate probabilities; [83, 84]. The numerical method therein outputs values that can be as accurate as required on input, provided enough time can be spent in the calculation. Since the numerical solution of (4.1.1) requires evaluating the constraint c (and a gradient) at many trial points x , the evaluation is not done exactly, but with some error, whose sign is unknown. As a result, and in spite of convexity, a linearization of the form $c_x + \langle g_{c_x}, \cdot - x \rangle$ with $c_x \approx c(x)$ and g_{c_x} an approximate subgradient, may locally overestimate the function $c(\cdot)$. An oracle providing such linearizations will be referred to as an upper inexact oracle. This naming is somewhat unsatisfactory because the linearizations need not overestimate the function c at all or would only do so locally. A more verbose choice would have been locally upper inexact oracle. This choice is also somewhat unsatisfactory as the provided linearization may overestimate c on a rather large set. We will therefore speak of *upper* inexact oracles.

To circumvent this difficulty, in this section we present a bundle method specially tailored to solve problems of the form (4.1.1) when computing f and/or c (as well as respective gradients) is computationally heavy. The algorithm is special because it solves a constrained non-smooth problem based on the information provided by an inexact oracle, possibly of upper type. This means that the oracle output provides linearizations for f and c in (4.1.1) that are inexact and may locally overestimate the corresponding function. The simpler case of lower oracles, yielding linearizations that always remain below the convex function, is also considered as a corollary. The convergence analysis of bundle methods with lower oracles is simpler, because it fits better the usual exact framework, in which the oracle linearizations define cutting planes for the function of interest (f and c in our case).

For unconstrained problems, bundle methods dealing with inexact oracles can be found in [113, 219, 128, 68, 70, 46]. Most of these works consider only lower oracles; we refer to [46] for a discussion on how such a setting considerably simplifies the convergence analysis. For constrained problems like (4.1.1), inexact bundle methods are more rare; see [129, 132, 44]. These works consider oracles that are either lower ones, or asymptotically exact. In this section, we give a method suitable for the more general upper setting, and hence, adapted to the hydro application of interest.

This section is organized as follows. Section 4.1.1 is devoted to bundle methods for upper inexact oracles. After giving the initial bundle setting in § 4.1.1.1, the attenuation step required for the method to converge in the presence of oracle noise is explained in § 4.1.1.2. The new bundle algorithm is given in full details in § 4.1.1.3. Based on Section 4.1.2 asymptotic results, Section 4.1.3 proves convergence of the method to a solution of (4.1.1), up to the accuracy provided by the oracle. In particular, we show that for lower oracles that are asymptotically exact, the method finds an exact minimizer whenever

(4.1.1) has a Slater point. Since our setting is more general than previous work (cf. the discussion in § 4.1.4.1), our approach significantly generalizes and extends results in the literature. The specific unit-commitment application is considered in the last two sections. Section 4.1.5 describes the hydro valley considered for the numerical tests, formulates the joint chance constrained problem to be solved, and discusses the upper inexact oracle employed. The different solvers used for comparison and a thorough set of numerical tests showing the interest of the approach are given in Section 4.1.6.

4.1.1 Designing Bundle methods for Constrained Optimization

Following the lead of [211], the numerical solution of (4.1.1) is addressed by means of an improvement function $H_\tau : \mathbb{R}^n \rightarrow \mathbb{R}$ defined by

$$H_\tau(y) = \max\left(f(y) - \tau_1, c(y) - \tau_2\right), \text{ for suitably chosen scalar targets } \tau_1, \tau_2. \quad (4.1.2)$$

However, unlike the exact setting considered in [211], the oracles which provide function and subgradient values for f and c make calculations with some error. For this reason, our method minimizes approximations of H_τ , built using oracle information computed with some inaccuracy. For clarity, at any point $x \in \mathbb{R}^n$ the symbols $f(x)$ and $c(x)$ refer to exact function values. Following [128], to denote inexact values the argument is put as a subscript, like in f_x, c_x (for functions) and g_{f_x} and g_{c_x} (for subgradients). Accordingly, given an iterate $x^j \in X$, the oracle provides f_{x^j} and $g_{f_{x^j}}$ shortened for convenience to $f^j = f_{x^j}$ and $g_f^j = g_{f_{x^j}}$, and similarly for the c -values.

4.1.1.1 Initial setting

We assume that at any $x^j \in X$, the oracle provides

$$\begin{aligned} f^j \quad \text{and} \quad c^j, & \quad \text{estimates for the functional values, as well as} \\ g_f^j \quad \text{and} \quad g_c^j, & \quad \text{estimates for the respective subgradients.} \end{aligned} \quad (4.1.3)$$

Since in this oracle the signs of the errors, e.g., $f(x^j) - f^j$, are not specified, the true function values can be either overestimated or underestimated, and similarly for the subgradients. In particular, nothing is known on the linearizations, e.g., $f^j + \langle g_f^j, \cdot - x^j \rangle$, that may locally overestimate the corresponding function, e.g., f . Further conditions on the (possibly upper) inexact oracle will be required in what follows, as needed (cf. (4.1.21) and (4.1.31) below).

Along the iterations the method keeps aside a reference solution called the stability center. The current stability center is denoted by \hat{x}^k at iteration k and has function values denoted by \hat{f}^k and \hat{c}^k . This center corresponds to some past iterate that was singled out because it produced significant progress towards the goal of solving (4.1.1). Progress is measured with respect to the current approximation of the improvement function, in a sense to be made clear below.

Specifically, the k -th inexact improvement function is

$$h_y^k = \max\left(f_y - \tau_1^k, c_y - \tau_2^k\right) \quad \text{where} \quad \begin{cases} \tau_1^k = \hat{f}^k + \rho_k \max(\hat{c}^k, 0) & \text{for } \rho_k \geq 0 \\ \tau_2^k = \sigma_k \max(\hat{c}^k, 0) & \text{for } \sigma_k \geq 0. \end{cases} \quad (4.1.4)$$

In the expression above, the targets τ^k are more general than those considered in [211], which correspond to taking null penalties ρ_k and σ_k . Relations with other improvement functions in the literature that are also covered by the setting (4.1.4) are discussed in § 4.1.4.1.

The oracle output is collected along iterations to form the Bundle of information

$$\mathcal{B}^k = \{\hat{x}^k, \hat{f}^k, \hat{c}^k\} \cup \{(x^j, f^j, c^j, g_{\mathbf{f}}^j, g_{\mathbf{c}}^j) : j \in J^k\} \quad \text{for } J^k \subset \{1, \dots, k\}.$$

Having this information, the k -th inexact improvement function is modelled by a convex function $\mathcal{M}_k : \mathbb{R}^n \rightarrow \mathbb{R}$ which uses approximate cutting-plane functions \check{f}_k and \check{c}_k :

$$\mathcal{M}_k(y) = \max\left(\check{f}_k(y) - \tau_1^k, \check{c}_k(y) - \tau_2^k\right) \quad \text{where} \quad \begin{cases} \check{f}_k(y) = \max\{f^j + \langle g_{\mathbf{f}}^j, y - x^j \rangle : j \in J^k\} \\ \check{c}_k(y) = \max\{c^j + \langle g_{\mathbf{c}}^j, y - x^j \rangle : j \in J^k\}. \end{cases} \quad (4.1.5)$$

To generate iterates, the algorithm chooses a prox-parameter $\mu_k > 0$ and solves the quadratic program (QP)

$$x^{k+1} = \arg \min_{y \in X} \left\{ \mathcal{M}_k(y) + \frac{1}{2} \mu_k \|y - \hat{x}^k\|^2 \right\}. \quad (4.1.6)$$

As a result, $x^{k+1} \in X$ and

$$x^{k+1} = \hat{x}^k - \frac{1}{\mu^k} (G^k + \nu^k) \quad \text{where} \quad G^k \in \partial \mathcal{M}_k(x^{k+1}) \text{ and } \nu^k \in N_X(x^{k+1}), \quad (4.1.7)$$

where $N_X(x^{k+1})$ denotes the normal cone (of convex analysis) of X at the new iterate and $\partial \mathcal{M}_k(x^{k+1})$ the sub-gradient of \mathcal{M}_k at x^{k+1} . After solving the problem, the *aggregate linearization*

$$\mathbf{M}^k(y) = \mathcal{M}_k(x^{k+1}) + \langle G^k, y - x^{k+1} \rangle, \quad (4.1.8)$$

which is an affine function, can be defined. Clearly, because $G^k \in \partial \mathcal{M}_k(x^{k+1})$,

$$\mathbf{M}^k(y) \leq \mathcal{M}_k(y) \quad \text{for all } y \in \mathbb{R}^n. \quad (4.1.9)$$

The last ingredient in the bundle method is given by the *aggregate error*, defined by

$$\mathbf{E}^k = h_{\hat{x}^k}^k - \mathbf{M}^k(\hat{x}^k) - \langle \nu^k, \hat{x}^k - x^{k+1} \rangle. \quad (4.1.10)$$

In view of (4.1.8), for any y it holds that $G^k = \nabla \mathbf{M}^k(y)$. Therefore, because $\mathbf{M}^k(\hat{x}^k) = \mathbf{M}^k(y) + \langle G^k, \hat{x}^k - y \rangle$ with $\mathbf{M}^k \leq \mathcal{M}_k$ by (4.1.9), we derive the relation

$$h_{\hat{x}^k}^k - \mathbf{E}^k \leq \mathcal{M}_k(y) + \langle G^k + \nu^k, \hat{x}^k - y \rangle \quad \text{for all } y \in X, \quad (4.1.11)$$

where we used the fact that the term $\langle \nu^k, x^{k+1} - y \rangle$ is nonnegative for all $y \in X$, because $\nu^k \in N_X(x^{k+1})$.

4.1.1.2 Handling inexact oracle information

Usually, the noise introduced by the inexact evaluations is deemed “too large” when the function value at the algorithmic center is below the minimum model value (a situation that is impossible with an exact oracle, by convexity). For our setting, this amounts to checking if the noise measurement quantity defined below is negative:

$$h_{\hat{x}^k}^k - \left(\mathcal{M}_k(x^{k+1}) + \frac{1}{2}\mu_k \|x^{k+1} - \hat{x}^k\|^2 \right) < 0.$$

When the relation above holds, the algorithm maintains the model and the center, and reduces the prox-parameter. The new iterate yields a smaller noise measurement quantity, thus attenuating the noise induced by the inexact bundle information. For the sake of numerical versatility, we consider here an alternative mechanism that is more general, and checks asymptotic satisfaction of the inequality above, based on a relative criterion. More precisely, noise is considered too large if

$$\frac{h_{\hat{x}^k}^k - \left(\mathcal{M}_k(x^{k+1}) + \frac{1}{2}\mu_k \|x^{k+1} - \hat{x}^k\|^2 \right)}{\frac{1}{2}\mu_k \|x^{k+1} - \hat{x}^k\|^2} < -\beta_k \quad (4.1.12)$$

for a parameter β_k satisfying (4.1.15) below.

To measure progress towards the goal of solving (4.1.1), certain predicted decrease δ^k is employed. Usual definitions for the decrease are $\delta^k = h_{\hat{x}^k}^k - \mathcal{M}_k(x^{k+1})$, or $\delta^k = h_{\hat{x}^k}^k - \mathcal{M}_k(x^{k+1}) - \frac{1}{2}\mu_k \|x^{k+1} - \hat{x}^k\|^2$. We consider a slightly more general variant, and let

$$\delta^k = h_{\hat{x}^k}^k - \mathcal{M}_k(x^{k+1}) - \frac{1}{2}\alpha_k\mu_k \|x^{k+1} - \hat{x}^k\|^2, \quad (4.1.13)$$

for a parameter α_k satisfying (4.1.15) below.

Since the numerator in (4.1.12) equals $\delta^k - \frac{1}{2}(1-\alpha_k)\mu_k \|x^{k+1} - \hat{x}^k\|^2$, we see that detecting the need of a noise attenuation step amounts to checking satisfaction of the inequality

$$\delta^k < \frac{1}{2}\left(1 - (\alpha_k + \beta_k)\right)\mu_k \|x^{k+1} - \hat{x}^k\|^2. \quad (4.1.14)$$

The choice of parameters α_k, β_k should ensure that the nominal decrease in (4.1.16) is nonnegative when noise is not too large. Accordingly, we suppose that

$$\exists b > -1 \text{ and } B > 0 \text{ such that } \beta_k \in [b, 1 - \alpha_k - B] \text{ for } \alpha_k \in [0, 2]. \quad (4.1.15)$$

Only when noise is declared acceptable, that is when (4.1.14) does not hold, the algorithm examines if the new iterate is good enough to become the next center by checking

$$\begin{cases} \text{either if } f^{k+1} \leq \hat{f}^k - m\delta^k & \text{and } c^{k+1} \leq 0 & \text{when } \hat{c}^k \leq 0, \\ \text{or if } c^{k+1} \leq \hat{c}^k - m\delta^k & & \text{when } \hat{c}^k > 0. \end{cases} \quad (4.1.16)$$

When the relation holds, the iteration is declared serious, because it provides a new algorithmic center: $\hat{x}^{k+1} = x^{k+1}$. Otherwise, the center is maintained and the iteration is declared null.

The rationale behind (4.1.16) is to measure progress towards minimization of (4.1.1) by focusing either in reducing the objective value without losing feasibility if the center is approximately feasible. Otherwise, when $\hat{c}^k > 0$, the emphasis is put in reducing infeasibility by checking the second condition in (4.1.16).

The parameters α_k, β_k in the criterion (4.1.14) make it possible to control the relation between noise attenuation and descent, a flexibility that can help the numerical performance of the algorithm. More specifically, to progress towards a solution, it is preferable for the algorithm to:

- make more serious iterations, because serious iterates converge to a solution, and
- have few noise attenuation steps. Noise attenuation steps are undesirable to occur often, because they prevent the algorithm from having “true” iterates, for which to check the descent condition.
- However, checking (4.1.14) does not involve any f/c -oracle calculation at x^{k+1} , and can therefore be considered an inexpensive test.

The flexibility introduced by the additional parameters α_k, β_k allows the user to seek for a trade-off between the time spent in oracle calculations and the CPU time required for the algorithm to find a solution to (4.1.1). By (4.1.16), more serious iterations are achieved by taking larger α_k 's (yielding smaller δ^k 's), while a larger β_k reduces the left handside term in (4.1.14), making less likely noise attenuation. Our numerical experience in Section 4.1.6 shows how different choices of these parameters impact the numerical performance, both in terms of CPU time and accuracy.

We now list some consequences resulting from the various definitions above, obtained with some simple algebraic manipulations.

First, by (4.1.8) written with $y = \hat{x}^k$ and (4.1.13),

$$h_{\hat{x}^k}^k - \mathbf{M}^k(\hat{x}^k) + \langle G^k, \hat{x}^k - x^{k+1} \rangle = \delta^k + \frac{1}{2} \alpha_k \mu_k \|x^{k+1} - \hat{x}^k\|^2 .$$

Together with (4.1.10) and (4.1.7) we see that

$$\mathbf{E}^k = \delta^k - \frac{2 - \alpha_k}{2} \mu_k \|x^{k+1} - \hat{x}^k\|^2 \quad \text{and, therefore,} \quad \mathbf{E}^k \leq \delta^k \quad \text{at all iterations} \quad (4.1.17)$$

because $\alpha_k \leq 2$, by (4.1.15).

Second, by adding δ^k to both sides of the identity (4.1.17), we see that

$$\text{inequality (4.1.14) holds} \quad \iff \quad \delta^k + \mathbf{E}^k < -\frac{(\alpha_k + 2\beta_k)}{2} \mu_k \|x^{k+1} - \hat{x}^k\|^2 \quad (4.1.18)$$

4.1.1.3 A Non-smooth Optimization Solver for Inexact Oracles

We now give our bundle algorithm for solving problem (4.1.1).

Algorithm 4.1 (Proximal Bundle). *We assume given an oracle computing approximate f/c values as in (4.1.3) for any $x \in X$, possibly of upper type.*

Step 0 (Input and Initialization) *Select a initial starting point \hat{x}^0 a stopping tolerance $\text{TOL} \geq 0$, an Armijo-like parameter $m \in (0, 1)$. Initialize the iteration counter $k = 0$, the bundle index set $J^0 := \{0\}$, and the first candidate point $x^0 := \hat{x}^0$. Call (4.1.3) to compute f^0, c^0 as well as $g_{\mathbf{f}}^0$ and $g_{\mathbf{c}}^0$. Choose the starting prox-parameter $\mu_0 > 0$, parameters α_0, β_0 satisfying (4.1.15), and penalties $\rho_0, \sigma_0 \geq 0$ satisfying (4.1.20) below.*

Step 1 (Model Generation and QP Subproblem) *Having the current algorithmic center \hat{x}^k , the bundle \mathcal{B}^k , the prox-parameter μ_k and the penalties ρ_k, σ_k , define the convex piecewise linear model function \mathcal{M}_k and compute $x^{k+1} = \arg \min\{\mathcal{M}_k(y) + \frac{1}{2}\mu_k \|y - \hat{x}^k\|^2 : y \in X\}$. Define the predicted decrease δ^{k+1} as in (4.1.13).*

Step 2 (Noise attenuation test)

If condition (4.1.14) is true, noise is too large: decrease the prox-parameter as in (4.1.19b) below; maintain the center, the bundle, and the penalties:

$$\left(\hat{x}^{k+1}, \mathcal{B}^{k+1}, \rho_{k+1}, \sigma_{k+1}\right) = \left(\hat{x}^k, \mathcal{B}^k, \rho_k, \sigma_k\right);$$

choose parameters $\alpha_{k+1}, \beta_{k+1}$ satisfying (4.1.15), and loop to Step 1.

Otherwise, if (4.1.14) does not hold, proceed to Step 3.

Step 3 (Stopping Test and New Oracle Information) *If $\delta^{k+1} \leq \text{TOL}$ then stop. Otherwise call the oracle to obtain $f^{k+1}, c^{k+1}, g_{\mathbf{f}}^{k+1}$ and $g_{\mathbf{c}}^{k+1}$.*

Step 4 (Serious step test) *Check the descent condition (4.1.16). If this condition is true, declare a serious iteration and set $\hat{x}^{k+1} = x^{k+1}$. Otherwise, declare a null step and maintain the center: $\hat{x}^{k+1} = \hat{x}^k$.*

Step 5 (Bundle Management and updates) *Choose a new prox-parameter μ_{k+1} satisfying (4.1.19a) if the iteration was declared serious or satisfying (4.1.19c) whenever the iteration was declared null. In all cases choose parameters $\alpha_{k+1}, \beta_{k+1}$ satisfying (4.1.15) and penalties ρ_{k+1}, σ_{k+1} satisfying (4.1.20) below. Define the new bundle \mathcal{B}^{k+1} , for example by appending to the index set the last iterate information: $J^{k+1} = J^k \cup \{k+1\}$. Increase k by 1 and loop to Step 1.*

Both in Step 2 and Step 5 there is some freedom in the choice of the new bundle \mathcal{B}^{k+1} . When noise is excessive, as in Step 2, the conservative choice of keeping the same cutting-plane models for both f and c seems reasonable. Alternative choices for managing the bundle in Step 5 are discussed after Lemmas 4.1.2, 4.1.3, and 4.1.4, for the cases of serious, noisy, and null iterations, respectively.

We now explain how Algorithm 4.1 handles the update of its prox-parameter μ_k and penalties ρ_k, σ_k . For the prox-parameter, the update uses positive constants μ_{\max} and

Δ , as follows:

$$\mu_{k+1} \leq \mu_{\max} < +\infty \quad \text{if iteration } k \text{ was declared serious} \quad (4.1.19a)$$

$$\mu_{k+1} \leq \mu_k - \Delta < \mu_k \quad \text{if iteration } k \text{ resulted in noise attenuation} \quad (4.1.19b)$$

$$\mu_{k+1} \in [\mu_k, \mu_{\max}] \quad \text{if iteration } k \text{ was declared null.} \quad (4.1.19c)$$

The rule for the penalty parameters uses a positive constant RS and is given below:

$$0 \leq \sigma_{k+1} \leq 1 \text{ and } \rho_{k+1} \geq 0 \text{ satisfy } 1 - \sigma_{k+1} + \rho_{k+1} \geq RS > 0. \quad (4.1.20)$$

The main purpose of these conditions is to ensure satisfaction of the relations stated in the proposition below.

Proposition 4.1.1 (Consequences of penalization updates). *When Algorithm 4.1 uses the rule (4.1.20) for the penalties, the following holds.*

– At every iteration k ,

$$h_{\hat{x}^k}^k = 0 \quad \text{if } \hat{c}^k \leq 0 \quad \text{and} \quad h_{\hat{x}^k}^k = \hat{c}^k(1 - \sigma_k) \geq 0 \quad \text{otherwise.}$$

– At every iteration k declared null, the inequality $h_{x^{k+1}}^k > h_{\hat{x}^k}^k - m\delta^k$ is satisfied.

– Suppose that, in addition, the oracle (4.1.3) is bounded, in the sense that

$$\text{the inexact values } \{f^k, c^k\} \text{ and } \{\|g_{\hat{x}}^k\|, \|g_c^k\|\} \quad (4.1.21)$$

are bounded for every sequence $\{x^k\} \subset X$. Then, there exist positive constants M and M' such that for any iteration k that is declared null or needing noise attenuation

$$\mathcal{M}_k(\hat{x}^k) \leq \max(M - \hat{f}^k, M) \quad \text{and} \quad h_{\hat{x}^k}^k \leq M'. \quad (4.1.22)$$

Proof. From (4.1.4), it readily follows that $\hat{c}^k \leq 0$ implies $\tau_1^k = \hat{f}^k$, whereas $\hat{c}^k > 0$ implies $\tau_1^k = \hat{f}^k + \rho_k \hat{c}^k$ and $\tau_2^k = \sigma_k \hat{c}^k$. As a consequence $\hat{c}^k \leq 0$ implies $h_{\hat{x}^k}^k = 0$. Inversely when $\hat{c}^k > 0$, we have $h_{\hat{x}^k}^k = \hat{c}^k \max(-\rho_k, 1 - \sigma_k) = \hat{c}^k(1 - \sigma_k) \geq 0$ because $\sigma_k \in [0, 1]$ and $\rho_k \geq 0$ by (4.1.20).

To prove the second item, let us first assume $\hat{c}^k > 0$. Then negating (4.1.16) and using the above derived identity for $h_{\hat{x}^k}^k$ gives

$$h_{x^{k+1}}^k = \max(f^{k+1} - \hat{f}^k - \rho_k \hat{c}^k, c^{k+1} - \sigma_k \hat{c}^k) \geq c^{k+1} - \sigma_k \hat{c}^k > \hat{c}^k(1 - \sigma_k) - m\delta^k = h_{\hat{x}^k}^k - m\delta^k.$$

When $\hat{c}^k \leq 0$, negating (4.1.16) implies that either $f^{k+1} > \hat{f}^k - m\delta^k$ or $c^{k+1} > 0$. In the first case we establish:

$$h_{x^{k+1}}^k = \max(f^{k+1} - \hat{f}^k, c^{k+1}) \geq f^{k+1} - \hat{f}^k > -m\delta^k,$$

as was to be shown, since $h_{\hat{x}^k}^k = 0$. In the second case, since k is a null step, (4.1.14) does not hold and $\delta^k > 0$ as a consequence. We thus establish $h_{x^{k+1}}^k = \max(f^{k+1} - \hat{f}^k, c^{k+1}) \geq c^{k+1} > 0 \geq h_{\hat{x}^k}^k - m\delta^k$ as was to be shown.

Finally, to see (4.1.22), notice that $x^k \in X$ with X bounded, so (4.1.21) ensures that each linearization $f^k + \langle g_{\mathbf{f}}^k, \cdot - x^k \rangle$ (or its c -counterpart) is bounded over X . In particular, both $\check{f}_k(\hat{x}^k)$ and $\check{c}_k(\hat{x}^k)$ are bounded by some constant M . In view of the model definition (4.1.5) and (4.1.4), $\mathcal{M}_k(\hat{x}^k) \leq \max(M - \tau_1^k, M - \tau_2^k) = \max(M - \hat{f}^k - \rho_k \max(\hat{c}^k, 0), M - \sigma_k \max(\hat{c}^k, 0))$. The bound for $\mathcal{M}_k(\hat{x}^k)$ follows, because the penalty terms are nonnegative by (4.1.20). An analogous reasoning gives the bound for $h_{\hat{x}^k}^k$. \square

4.1.2 Asymptotic Analysis

We now analyse the different cases that can arise when the algorithm in Section 4.1 loops forever, i.e., $k \rightarrow \infty$. Then only one of the following mutually exclusive cases can occur:

- either there are infinitely many serious iterates,
- or the stability center remains unchanged after a finite number of iterations. In this case,
 - either there is an infinite number of noise attenuation steps,
 - or there is a finite number of noise attenuation steps and eventually only null steps are done.

The first case is considered in Lemma 4.1.2, the second case in Lemma 4.1.3 and the last case in Lemma 4.1.4.

Lemma 4.1.2 (Infinitely many serious iterations). *Consider solving (4.1.1) with Algorithm 4.1 using an oracle satisfying (4.1.3) and (4.1.21), with parameters α_k, β_k satisfying (4.1.15).*

Let K_s denote the set gathering indices of serious iterations. If there are infinitely many of such indices, and the prox-parameter sequence satisfies (4.1.19a), then

$$h_{\hat{x}^k}^k \leq \mathcal{M}_k(y) + o(1/k) \quad \text{for all } y \in X \text{ and } k \in K_s \text{ sufficiently large.}$$

Proof. Consider $k \in K_s$. We first show that $\delta^k \rightarrow 0$. If for all serious steps we have $c^{k+1} > 0$ then (4.1.16) implies that

$$\hat{c}^{k+1} = c^{k+1} \leq c^k - m\delta^k = \hat{c}^k - m\delta^k, \quad (4.1.23)$$

noting that $\delta^k \geq 0$, because serious steps can only take place when no noise attenuation occurs. Since the nonincreasing sequence $\{\hat{c}^k\}_{K_s}$ is bounded below by 0, it converges. From (4.1.23) we deduce that $0 \leq \delta^k \leq \frac{c^k - c^{k+1}}{m}$. Since $\{\hat{c}^k\}_{K_s}$ converges this gives that $\delta^k \rightarrow 0$ as $K_s \ni k \rightarrow \infty$.

Otherwise, there is some $\bar{k} \in K_s$ such that $\hat{c}_{\bar{k}} \leq 0$. In view of (4.1.16), all subsequent serious iterates are feasible: for each serious step $\bar{k} \leq k \in K_s$ we have $\hat{c}^{k+1} = c^{k+1} \leq 0$ and

$$\hat{f}^{k+1} \leq \hat{f}^k - m\delta^k \quad \text{with } \delta^k \geq 0. \quad (4.1.24)$$

Thus, the nonincreasing sequence $\{\hat{f}^k\}_{\hat{k} \leq k \in K_s}$ is either unbounded below or converges. The first case is ruled out by (4.1.21). As for the second case, it implies that $\delta^k \rightarrow 0$ since $0 \leq \delta^k \leq \frac{f^k - f^{k+1}}{m}$.

We now show that both $G^k + \nu^k \rightarrow 0$ and $\mathbf{E}^k \rightarrow 0$. Since $(1 - (\alpha_k + \beta_k)) \geq B > 0$ by (4.1.15), the negation of (4.1.14) and (4.1.7) imply that

$$\begin{aligned} 0 \leftarrow \delta^k &\geq \frac{1}{2} \left(1 - (\alpha_k + \beta_k)\right) \mu_k \|x^{k+1} - \hat{x}^k\|^2 = \frac{1}{2\mu_k} \left(1 - (\alpha_k + \beta_k)\right) \|G^k + \nu^k\|^2 \\ &\geq \frac{B}{2\mu_k} \|G^k + \nu^k\|^2 \geq \frac{B}{2\mu_{max}} \|G^k + \nu^k\|^2 \geq 0 \end{aligned}$$

where we used (4.1.19a) for the last inequality. As a result, both $\|G^k + \nu^k\|^2 / \mu_k \rightarrow 0$ and $G^k + \nu^k \rightarrow 0$ as $K_s \ni k \rightarrow \infty$.

Since $\alpha_k \geq 0$ and $\alpha_k + \beta_k \leq 1 - B$ by (4.1.15), we deduce that $-(\alpha_k + 2\beta_k)/2 = -(\alpha_k + \beta_k) + \alpha_k/2 \geq B - 1$. To show that the aggregate error goes to 0, pass to the limit in the negation of (4.1.18) and use the above estimate to see that

$$\lim_{k \in K_s} \mathbf{E}^k \geq (B - 1) \lim_{k \in K_s} \mu_k \|x^{k+1} - \hat{x}^k\|^2 .$$

Since by (4.1.7), $\mu_k \|x^{k+1} - \hat{x}^k\|^2 = \|G^k + \nu^k\|^2 / \mu_k$ and we have just shown this term vanishes asymptotically, the error limit is nonnegative. Then $\mathbf{E}^k \rightarrow 0$, because $\mathbf{E}^k \leq \delta^k$ by (4.1.17) and $\delta^k \rightarrow 0$.

The stated inequality holds follows from (4.1.11) by boundedness of X and the fact that both $G^k + \nu^k$ and $\mathbf{E}^k \rightarrow 0$. \square

The analysis above shows that Step 5 of Algorithm 4.1 can freely manage the bundle at serious iterations. Of course, a richer bundle yields better cutting-plane models for f and c , so having larger index-sets J^k should improve the speed of the method (keeping in mind that large sets J^k make the QP subproblem more difficult).

Notice also that in the inequality in Lemma 4.1.2 the remainder $o(1/k)$ corresponds in fact to both $G^k + \nu^k$ and \mathbf{E}^k going to zero. Similar relations will hold when there is a finite number of serious iterations, as shown below.

In the next two cases, eventually no more serious steps occur and after a finite number of iterations the algorithmic center remains unchanged. As a result, there exists \hat{k} and \hat{x} such that $\hat{x}^k = \hat{x}$ for all $k > \hat{k}$. Unlike Lemma 4.1.2, the results below rely on conditions (4.1.20), required for the penalty parameters defining the inexact improvement function (4.1.4).

Lemma 4.1.3 (Infinitely many noisy iterations). *Consider solving (4.1.1) with Algorithm 4.1 using an oracle satisfying (4.1.3) and (4.1.21), with parameters α_k, β_k satisfying (4.1.15) and penalties as in (4.1.20).*

Suppose at iteration \hat{k} there is a last serious iterate \hat{x} and let K_a denote the set gathering indices of iterations larger than \hat{k} for which (4.1.14) holds and noise is deemed too large.

If there are infinitely many of such indices and (4.1.19b) holds then

$$h_{\hat{x}^k}^k = h_{\hat{x}}^k \leq \mathcal{M}_k(y) + o(1/k) \quad \text{for all } y \in X \text{ and } k \in K_a \text{ sufficiently large.}$$

Proof. Consider $k \in K_a$. By (4.1.17) and (4.1.14) we obtain the following estimate

$$2\mathbf{E}^k = 2\delta^k - (2 - \alpha_k)\mu_k \|x^{k+1} - \hat{x}^k\|^2 < -(1 + \beta_k)\mu_k \|x^{k+1} - \hat{x}^k\|^2 \leq 0, \quad (4.1.25)$$

since $\beta_k \geq b > -1$ by (4.1.15). Since $\mathbf{E}^k \leq 0$, $h_{\hat{x}^k}^k - \mathbf{E}^k \geq h_{\hat{x}^k}^k$, and with (4.1.11) we see that

$$h_{\hat{x}^k}^k \leq \mathcal{M}_k(y) + \langle G^k + \nu^k, \hat{x} - y \rangle \quad \text{for all } y \in X.$$

Since the set X is bounded, the stated inequality would hold if $G^k + \nu^k \rightarrow 0$. To show this result, first note that (4.1.10) and (4.1.8) imply that

$$-\mathbf{E}^k = \mathcal{M}_k(x^{k+1}) + \langle G^k + \nu^k, \hat{x} - x^{k+1} \rangle - h_{\hat{x}^k}^k \leq \mathcal{M}_k(\hat{x}) - h_{\hat{x}^k}^k,$$

where the inequality comes from (4.1.7). Let \hat{f} denote the f -value computed by the oracle at \hat{x} . By the first item in Proposition 4.1.1, $h_{\hat{x}^k}^k \geq 0$ because $\sigma_k \leq 1$ by (4.1.20), so $-\mathbf{E}^k \leq \mathcal{M}_k$. Since $\mathcal{M}_k(\hat{x}) \leq \max(\hat{f} - M, \hat{f})$ by the third item in the proposition, for some constant \hat{M}

$$-\mathbf{E}^k \leq \hat{M} \text{ for all } k \in K_a.$$

From (4.1.25) and the condition $\beta_k \geq b$ from (4.1.15) we obtain the inequality $\mathbf{E}^k < -\frac{1}{2}(1 + b)\mu_k \|x^{k+1} - \hat{x}^k\|^2$. Moreover, using (4.1.7) and the fact that $1 + b > 0$ by (4.1.15), this means that

$$\frac{2}{1 + b}\mu_k \mathbf{E}^k < -\|G^k + \nu^k\|^2$$

or, equivalently, that

$$0 \leq \|G^k + \nu^k\| \leq \sqrt{-\frac{2}{1 + b}\mu_k \mathbf{E}^k} \leq \sqrt{\frac{2}{1 + b}\mu_k \hat{M}}.$$

Since the update in (4.1.19b) ensures that the sequence $\{\mu_k\}_{k \in K_a}$ is strictly decreasing, when $K_a \ni k \rightarrow \infty$ we obtain that $\mu_k \rightarrow 0$ and $G^k + \nu^k \rightarrow 0$, as desired. \square

For the result above to hold, Step 2 in Algorithm 4.1 only needs the sequence $\{\mu_k\}_{K_a}$ to be strictly decreasing, and the left bound in (4.1.22) to hold. So, as for the case of infinitely many serious iterations, there is freedom in how the bundle is managed when noise is deemed too large. Since in this case there is no new oracle information, it seems reasonable to maintain the current bundle.

In a manner similar to Lemma 4.1.2, the remainder term in Lemma 4.1.3 corresponds in fact to $G^k + \nu^k \rightarrow 0$ with $\mathbf{E}^k \leq 0$.

The final result considers that there are finitely many serious and noise attenuation steps, which implies that the algorithm makes infinitely many consecutive null steps. For this

case, the model is required to satisfy the following conditions whenever the iteration k is declared null:

$$\mathcal{M}_{k+1}(y) \geq \mathbf{M}^k(y) \text{ and} \quad (4.1.26a)$$

$$\mathcal{M}_{k+1}(y) \geq \max\left(f^{k+1} + \langle g_{\mathbf{f}}^{k+1}, y - x^{k+1} \rangle - \tau_1^k, c^{k+1} + \langle g_c^{k+1}, y - x^{k+1} \rangle - \tau_2^k\right) \quad (4.1.26b)$$

The result below uses standard arguments in bundle methods [39], taking advantage of the boundedness relations in (4.1.21) and (4.1.22) to adapt those arguments to the inexact improvement function setting.

Lemma 4.1.4 (Infinitely many consecutive null iteration). *Consider solving (4.1.1) with Algorithm 4.1 using an oracle satisfying (4.1.3) and (4.1.21), with parameters α_k , β_k satisfying (4.1.15) and penalties as in (4.1.20).*

Suppose at iteration \hat{k} there is a last serious step, denoted by \hat{x} and after an iteration $\bar{k} > \hat{k}$ there are no more noise attenuation steps: eventually only null steps occur. Let K_n denote the set gathering indices of iterations larger than \bar{k} . If there are infinitely many of such indices and both (4.1.19c) and (4.1.26) hold, then

$$h_{\hat{x}^k}^k = h_{\hat{x}}^k \leq \mathcal{M}_k(y) + o(1/k) \text{ for all } y \in X \text{ and } k \in K_n \text{ sufficiently large.}$$

Furthermore, $x^k \rightarrow \hat{x}$ as $K_n \ni k \rightarrow \infty$.

Proof. Consider $k \in K_n$. Once again, the stated inequality will follow from (4.1.11) by boundedness of X , if we show that both $G^k + \nu^k \rightarrow 0$ and $\mathbf{E}^k \rightarrow 0$. In turn, these results follow from showing that $\delta^k \rightarrow 0$.

To see that $\delta^k \rightarrow 0$, we start by expanding squares and using (4.1.7) to write the identity

$$\begin{aligned} 2\langle G^k + \nu^k, y - x^{k+1} \rangle &= 2\mu_k \langle \hat{x} - x^{k+1}, y - x^{k+1} \rangle \\ &= \mu_k \|x^{k+1} - \hat{x}\|^2 + \mu_k \|y - x^{k+1}\|^2 - \mu_k \|y - \hat{x}\|^2 \end{aligned} \quad (4.1.27)$$

for all $y \in \mathbb{R}^n$. Since the function \mathbf{M}^k is affine with gradient G^k , for all $y \in \mathbb{R}^n$

$$\begin{aligned} \mathbf{M}^k(y) &= \mathbf{M}^k(x^{k+1}) + \langle G^k, y - x^{k+1} \rangle \\ &= \mathbf{M}^k(x^{k+1}) + \langle G^k, y - x^{k+1} \rangle + \langle \nu^k, y - x^{k+1} \rangle - \langle \nu^k, y - x^{k+1} \rangle \quad (4.1.28) \\ &= \mathbf{M}^k(x^{k+1}) + \frac{1}{2}\mu_k \|x^{k+1} - \hat{x}\|^2 \\ &\quad + \frac{1}{2}\mu_k \|y - x^{k+1}\|^2 - \frac{1}{2}\mu_k \|y - \hat{x}\|^2 - \langle \nu^k, y - x^{k+1} \rangle \\ &= \text{OV}^k + \frac{1}{2}\mu_k \|y - x^{k+1}\|^2 - \frac{1}{2}\mu_k \|y - \hat{x}\|^2 - \langle \nu^k, y - x^{k+1} \rangle, \end{aligned}$$

where in the last equality we define $\text{OV}^k := \mathcal{M}_k(x^{k+1}) + \frac{1}{2}\mu_k \|x^{k+1} - \hat{x}\|^2$ and use the relation $\mathbf{M}^k(x^{k+1}) = \mathcal{M}_k(x^{k+1})$ from (4.1.8) (the value OV^k is the optimal value of the

QP subproblem (4.1.6) defining x^{k+1}). Since $\nu^k \in N_X(x^{k+1})$ and $y \in X$, the term $-\langle \nu^k, y - x^{k+1} \rangle \geq 0$ and, hence, we derive from (4.1.28) that

$$\forall y \in X \quad \mathbf{M}^k(y) + \frac{1}{2}\mu_k \|y - \hat{x}\|^2 \geq \mathbf{OV}^k + \frac{1}{2}\mu_k \|y - x^{k+1}\|^2. \quad (4.1.29)$$

By evaluating at $y = \hat{x} \in X$, using (4.1.9) and the third item in Proposition 4.1.1, there exists some constant \hat{M} such that

$$\mathbf{OV}^k + \frac{1}{2}\mu_k \|\hat{x} - x^{k+1}\|^2 \leq \mathbf{M}^k(\hat{x}) \leq \mathcal{M}_k(\hat{x}) \leq \hat{M}.$$

In particular, the sequence $\{\mathbf{OV}^k\}$ is bounded from above.

Assumption (4.1.26a) in (4.1.29) yields that

$$\mathcal{M}_{k+1}(y) + \frac{1}{2}\mu_k \|y - \hat{x}\|^2 \geq \mathbf{OV}^k + \frac{1}{2}\mu_k \|y - x^{k+1}\|^2$$

for all $y \in X$. Since $\mu_{k+1} \geq \mu_k$ by (4.1.19c), by evaluating the inequality above at $y = x^{k+2}$ we see that

$$\mathbf{OV}^{k+1} \geq \mathbf{OV}^k + \frac{1}{2}\mu_k \|x^{k+2} - x^{k+1}\|^2,$$

and, being bounded above, the non-decreasing sequence $\{\mathbf{OV}^k\}$ converges. Since the righthand side is larger than \mathbf{OV}^k , we conclude that

$$\mathbf{OV}^{k+1} - \left(\mathbf{OV}^k + \frac{1}{2}\mu_k \|x^{k+2} - x^{k+1}\|^2 \right) \rightarrow 0 \quad \text{and, furthermore,} \quad \|x^{k+2} - x^{k+1}\|^2 \rightarrow 0 \quad (4.1.30)$$

because $\mu_k \geq \mu_{k+1}$ by (4.1.19c).

Recall that condition (4.1.15) implies that $\delta_k \geq 0$ when (4.1.14) fails, i.e., no noise is detected. The descent test also fails, so by the second item in Proposition 4.1.1,

$$h_{x^{k+1}}^k > h_{\hat{x}^k}^k - m\delta^k = h_{\hat{x}}^k - m\delta^k.$$

Adding δ^k to both terms and using the definition in (4.1.13) we obtain that

$$\begin{aligned} 0 \leq (1-m)\delta^k &< \delta^k + h_{x^{k+1}}^k - h_{\hat{x}^k}^k \\ &= h_{x^{k+1}}^k - \mathcal{M}_k(x^{k+1}) - \frac{1}{2}\alpha_k\mu_k \|x^{k+1} - \hat{x}\|^2 \\ &\leq h_{x^{k+1}}^k - \mathcal{M}_k(x^{k+1}) \\ &= h_{x^{k+1}}^k - \mathcal{M}_{k+1}(x^{k+2}) + \mathcal{M}_{k+1}(x^{k+2}) - \mathcal{M}_k(x^{k+1}). \end{aligned}$$

Writing (4.1.26b) for $y = x^{k+2}$ and using the Cauchy-Schwarz inequality yields that

$$\mathcal{M}_{k+1}(x^{k+2}) \geq \max(f^{k+1} - \tau_1^k - \|g_{\mathbf{f}}^{k+1}\| \|x^{k+2} - x^{k+1}\|, c^{k+1} - \tau_2^k - \|g_{\mathbf{c}}^{k+1}\| \|x^{k+2} - x^{k+1}\|).$$

By (4.1.21), there exists a constant Γ such that $\|g_{\mathbf{f}}^{k+1}\|, \|g_{\mathbf{c}}^{k+1}\| \leq \Gamma$ and, hence,

$$\mathcal{M}_{k+1}(x^{k+2}) \geq \max(f^{k+1} - \tau_1^k, c^{k+1} - \tau_2^k) - \Gamma \|x^{k+2} - x^{k+1}\|.$$

The first righthand side term above equals $h_{x^{k+1}}^k$, by (4.1.4). Continuing and using the definition of OV^k and (4.1.19c),

$$\begin{aligned}
0 \leq (1-m)\delta^k &\leq \Gamma \|x^{k+2} - x^{k+1}\| + \mathcal{M}_{k+1}(x^{k+2}) - \mathcal{M}_k(x^{k+1}) \\
&= \Gamma \|x^{k+2} - x^{k+1}\| + \text{OV}^{k+1} - \frac{1}{2}\mu_{k+1} \|x^{k+2} - \hat{x}\|^2 \\
&\quad - \text{OV}^k + \frac{1}{2}\mu_k \|x^{k+1} - \hat{x}\|^2 \\
&= \Gamma \|x^{k+2} - x^{k+1}\| + \text{OV}^{k+1} - \text{OV}^k - \frac{1}{2}\mu_k \|x^{k+2} - \hat{x}\|^2 \\
&\quad + \frac{1}{2}\mu_k \|x^{k+1} - \hat{x}\|^2 \\
&= \Gamma \|x^{k+2} - x^{k+1}\| + \text{OV}^{k+1} - (\text{OV}^k + \frac{1}{2}\mu_k \|x^{k+2} - x^{k+1}\|^2) \\
&\quad + \frac{1}{2}\mu_k (\|x^{k+2} - x^{k+1}\|^2 - \|x^{k+2} - \hat{x}\|^2 + \|x^{k+1} - \hat{x}\|^2).
\end{aligned}$$

Following (4.1.27), we can observe that the last three terms equal $\mu_k \langle x^{k+2} - x^{k+1}, \hat{x} - x^{k+1} \rangle$. By (4.1.7), $\mu_k(\hat{x} - x^{k+1}) = G^k + \nu^k$ and, since $\nu^k \in N_X(x^{k+1})$ and $x^{k+2} \in X$,

$$\begin{aligned}
\mu_k \langle x^{k+2} - x^{k+1}, \hat{x} - x^{k+1} \rangle &= \langle x^{k+2} - x^{k+1}, G^k + \nu^k \rangle \leq \langle x^{k+2} - x^{k+1}, G^k \rangle \\
&\leq \|G^k\| \|x^{k+2} - x^{k+1}\|
\end{aligned}$$

Since $G^k \in \partial\mathcal{M}_k(x^{k+1}) \subset \text{conv}\{g_{\mathbf{f}}^j, g_{\mathbf{c}}^j : j \in J_k\}$, assumption (4.1.21) implies that $\|G^k\| \leq \Gamma$ and, hence,

$$0 \leq (1-m)\delta^k \leq 2\Gamma \|x^{k+2} - x^{k+1}\| + \text{OV}^{k+1} - (\text{OV}^k + \frac{1}{2}\mu_k \|x^{k+2} - x^{k+1}\|^2).$$

In view of (4.1.30), the right handside above tends to zero. Since $m \in (0, 1)$, $\delta^k \rightarrow 0$, as desired.

The negation of (4.1.14), condition $1 - (\alpha_k + \beta_k) \geq B > 0$ from (4.1.15), and the fact that $\mu_k \geq \mu_{\bar{k}+1}$ from (4.1.19c) imply that

$$0 \leftarrow \delta^k \geq \frac{1}{2}\mu_{\bar{k}+1}B \|x^{k+1} - \hat{x}\|^2 \geq 0,$$

so $x^{k+1} \rightarrow \hat{x}$, as stated. Since by (4.1.7) $G^k + \nu^k = \mu_k(\hat{x} - x^{k+1})$ and by (4.1.19c) $\mu_k \leq \mu_{\max}$, we obtain that

$$0 \leq \frac{1}{\mu_{\max}} \|G^k + \nu^k\| \leq \frac{1}{\mu_k} \|G^k + \nu^k\| = \|x^{k+1} - \hat{x}\| \rightarrow 0 \text{ and, hence, } G^k + \nu^k \rightarrow 0.$$

Finally, using in (4.1.17) that $\alpha_k \geq 0$ by (4.1.15) gives

$$0 \leftarrow \delta^k \geq \mathbf{E}^k = \delta^k - \frac{2 - \alpha_k}{2}\mu_k \|x^{k+1} - \hat{x}\|^2 \geq \delta^k - \mu_k \|x^{k+1} - \hat{x}\|^2 \geq \delta^k - \mu_{\max} \|x^{k+1} - \hat{x}\|^2.$$

Since both righthand side terms go to zero, so does \mathbf{E}^k and the proof is finished. \square

For the result above to hold, at null steps the bundle needs to be managed in a manner ensuring the relations (4.1.26). Condition (4.1.26b) holds if the last generated information enters the bundle, that is,

$$k + 1 \in J^{k+1} \quad \text{which is equivalent to having } (x^{k+1}, f^{k+1}, c^{k+1}, g_f^{k+1}, g_c^{k+1}) \in \mathcal{B}^{k+1}.$$

As for (4.1.26a), the inequality is typically ensured by introducing aggregate information in the bundle. In our improvement function setting, this can be done by splitting the aggregate information into their f and c parts. Specifically, in view of the model definition (4.1.5), there is a multiplier $\lambda^k \in [0, 1]$ such that

$$\mathcal{M}_k(x^{k+1}) = \lambda^k(\check{f}_k(x^{k+1}) - \tau_1^k) + (1 - \lambda^k)(\check{c}_k(x^{k+1}) - \tau_2^k),$$

and

$$G^k = \lambda^k G_f^k + (1 - \lambda^k) G_c^k \quad \text{with } G_f^k \in \partial \check{f}_k(x^{k+1}), G_c^k \in \partial \check{c}_k(x^{k+1}).$$

For (4.1.26a) to hold it is then enough to take

$$(x^{k+1}, \check{f}_k(x^{k+1}), \check{c}_k(x^{k+1}), G_f^k, G_c^k) \in \mathcal{B}^{k+1}.$$

Similar calculations can be derived for economic bundles which, in the information $(x^j, f^j, c^j, g_f^j, g_c^j)$, replace the knowledge of the vector x^j by two scalars, referred to as linearization errors for f and c at \hat{x}^k . We refer to [211] for more details.

Finally, like in the case of infinitely many serious steps, the remainder term corresponds to $G^k + \nu^k \rightarrow 0$ with $\mathbf{E}^k \rightarrow 0$. For this reason, instead of the stopping criteria in Step 3 of Algorithm 4.1, one could stop when $G^k + \nu^k$ is sufficiently small, as long as \mathbf{E}^k is also small (serious and null cases) or nonpositive (noisy case).

4.1.3 Link with the Original Problem

In Section 4.1.2 we established the limiting behaviour of Algorithm 4.1. We still need to analyze in which sense the method converges to an approximate solution to problem (4.1.1). It is at this stage that the oracle errors play a major role. Our results below show what can be expected in terms of solving (4.1.1) for oracles that are of upper or lower type. Partly asymptotically exact and exact oracles are also considered.

4.1.4 Convergence Results

We start with a general result, suitable for oracles yielding inexact linearizations that may lie above the function. Specifically, we suppose that for any $x^k \in X$ the oracle output (4.1.3) is of upper type, in the sense that errors ε^k at iteration k and asymptotic error $\varepsilon \geq 0$ satisfy:

$$\forall y \in X \quad \begin{cases} f^k + \langle g_f^k, y - x^k \rangle \leq f(y) + \varepsilon^k \\ c^k + \langle g_c^k, y - x^k \rangle \leq c(y) + \varepsilon^k \end{cases} \quad \text{with } \varepsilon := \limsup \varepsilon^k. \quad (4.1.31)$$

Theorem 4.1.5 (Asymptotic Bounds for Upper Oracles). *Consider solving (4.1.1) with Algorithm 4.1 using an oracle (4.1.3) such that (4.1.21) and (4.1.31) hold, with parameters α_k, β_k satisfying (4.1.15) and penalties as in (4.1.20). Suppose the prox-parameter is updated according to (4.1.19), and the model satisfies (4.1.26).*

When the algorithm loops forever, for any accumulation point \bar{x} of the (bounded) sequence $\{\hat{x}^k\}$ and its limiting inexact f/c -values and parameters $(\bar{f}, \bar{c}, \bar{\rho}, \bar{\sigma})$ it holds that

$$\forall y \in X \quad \max(\bar{c}, 0)(1 - \bar{\sigma}) \leq \max\left(f(y) - \bar{f} - \bar{\rho} \max(\bar{c}, 0), c(y) - \bar{\sigma} \max(\bar{c}, 0)\right) + \varepsilon. \quad (4.1.32)$$

Moreover, :

(i) *If $\bar{c} > 0$ then*

$$\exists R : \bar{\rho} \geq R \implies \bar{c} \leq c(y) + \varepsilon \text{ for all } y \in X.$$

(ii) *If $\bar{c} \leq 0$ and the set $X_\varepsilon = \{y \in X : c(y) \leq -2\varepsilon\}$ is not empty, then*

$$\bar{f} \leq f(y) + \varepsilon \text{ for all } y \in X_\varepsilon.$$

Proof. When the algorithm loops forever, one of the index sets K_s, K_a, K_n , defined respectively in Lemmas 4.1.2, 4.1.3, 4.1.4, is infinite. Since $\hat{x}^k \in X$ and X is compact, for any of such sets there exists a subset K' such that $\{\hat{x}^k\}_{k \in K'} \rightarrow \bar{x}$, recalling that when $K' = K_a$ and $K' = K_n$ eventually $\hat{x}^k = \hat{x} = \bar{x}$ remains fixed. By the same three lemmas, and the first item in Proposition 4.1.1,

$$\max(\hat{c}^k, 0)(1 - \sigma_k) = h_{\hat{x}^k}^k \leq \mathcal{M}_k(y) + o(1/k) \text{ for all } y \in X \text{ and } k \in K' \text{ sufficiently large.}$$

By (4.1.31), $\mathcal{M}_k(y) \leq H_{\tau^k}(y) + \varepsilon^k$ for the exact improvement function (4.1.2) written with target $\tau = \tau^k$, so

$$\forall y \in X \quad \max(\hat{c}^k, 0)(1 - \sigma_k) \leq \max(f(y) - \tau_1^k, c(y) - \tau_2^k) + o(1/k) + \varepsilon^k. \quad (4.1.33)$$

By (4.1.4), the target is $\tau^k = (\hat{f}^k + \rho_k \max(\hat{c}^k, 0), \sigma_k \max(\hat{c}^k, 0))$ and by (4.1.21) the sequences $\{\hat{f}^k\}$ and $\{\hat{c}^k\}$ are bounded. Extracting from K' a further subsequence K if needed, we let

$$\bar{f} = \lim_{k \in K} \hat{f}^k, \quad \bar{c} = \lim_{k \in K} \hat{c}^k, \quad \bar{\rho} = \lim_{k \in K} \rho_k, \quad \bar{\sigma} = \lim_{k \in K} \sigma_k,$$

noting that $\bar{\rho}$ is not necessarily finite. Passing to the limit as $K \ni k \rightarrow \infty$ in (4.1.33) yields (4.1.32).

Consider first the case $\bar{c} > 0$. Since X is bounded and both f and c are real-valued, the constant $R = \frac{1}{\bar{c}} \left(\max_{y \in X} (f(y) - c(y)) - \bar{f} \right) + \bar{\sigma}$ is well defined and any $\bar{\rho} > R$ satisfies $(\bar{\rho} - \bar{\sigma})\bar{c} > f(y) - c(y) - \bar{f}$ for all $y \in X$. Since the inequality is equivalent to having $f(y) - \bar{f} - \bar{\rho} \max(\bar{c}, 0) < c(y) - \bar{\sigma} \max(\bar{c}, 0)$, the stated results follows from (4.1.32).

When $\bar{c} \leq 0$, (4.1.32) becomes $0 \leq \max\left(f(y) - \bar{f}, c(y)\right) + \varepsilon$ and evaluating at any $y \in X_\varepsilon$ gives the final result. \square

Theorem 4.1.5 states that, as long as ρ_k is managed as a penalty parameter (for instance $\rho_{k+1} = 2\rho_k$ if $c_{\hat{x}^{k+1}} > 0$, and $\rho_{k+1} = \rho_k$ otherwise), Algorithm 4.1 will eventually detect problem (4.1.1) as infeasible up to the accuracy ε , or it will find an approximate minimizer in the sense stated by the second item in the theorem. For the latter to happen, the accuracy should be small enough to ensure nonemptiness of the set X_ε (noting that in this set the factor -2 could be replaced by any value strictly smaller than -1).

We now state a refinement of this result, for the case when inaccurate linearizations eventually stay below the functions f and c . We refer to this situation as having “lower” oracles.

Corollary 4.1.6 (Convergence for lower oracles). *In the setting of Theorem 4.1.5, suppose $\varepsilon = 0$ in (4.1.31). If problem (4.1.1) has a Slater point and $\bar{\rho}$ is sufficiently large, then $\bar{c} \leq 0$ and $\bar{f} \leq f(y)$ for all y feasible in (4.1.1).*

Proof. The case $\bar{c} \leq 0$ is Theorem 4.1.5(ii) with $\varepsilon = 0$. As for the case $\bar{c} > 0$, it is excluded by observing that for $\bar{\rho}$ sufficiently large (4.1.32) becomes

$$\bar{c}(1 - \bar{\sigma}) \leq \max\left(f(y) - \bar{f} - \bar{\rho}\bar{c}, c(y) - \bar{\sigma}\bar{c}\right) = c(y) - \bar{\sigma}\bar{c},$$

so $0 < \bar{c} \leq c(y)$ for all $y \in X$, an inequality that cannot hold when y is the Slater point. \square

A further refinement is possible for lower oracles that are partly asymptotically exact, see [81], [131], [44]. Specifically, these are oracles that become progressively more and more accurate at serious steps. We now show that, unlike the previous cases, in this situation the penalty ρ_k does not need to increase to infinity when the center is deemed infeasible (that is, when $c_{\hat{x}^k} > 0$).

Corollary 4.1.7 (Convergence for partly asymptotically exact lower oracles). *In the setting of Theorem 4.1.5, suppose $\varepsilon = 0$ in (4.1.31) and calculations are eventually exact for serious steps: $f(\bar{x}) = \bar{f}$ and $c(\bar{x}) = \bar{c}$. The following holds:*

- (i) *Either (4.1.1) is feasible and $c(\bar{x}) \leq 0$ with \bar{x} solving (4.1.1).*
- (ii) *Or $c(\bar{x}) > 0$ and (4.1.1) is infeasible with \bar{x} minimizing infeasibility.*

As a result, when (4.1.1) has a Slater point, only item (i) is possible.

Proof. When $\bar{c} \leq 0$, the first result is Theorem 4.1.5(ii), recalling that $\bar{f} = f(\bar{x})$ and $\varepsilon = 0$.

Similarly for $\bar{c} > 0$ and $\bar{\rho} = +\infty$, applying the first item in Theorem 4.1.5 with $\bar{c} = c(\bar{x})$ and $\varepsilon = 0$. When $\bar{c} > 0$ and $\bar{\rho}$ is finite (not necessarily larger than R in the theorem), adding $\bar{\sigma}\bar{c}$ to both sides of (4.1.32) and using that $\varepsilon = 0$ together with the assumption that $c(\bar{x}) = \bar{c}$ and $f(\bar{x}) = \bar{f}$ gives that

$$c(\bar{x}) \leq H(y) := \max\left(f(y) - f(\bar{x}) - (\bar{\rho} - \bar{\sigma})c(\bar{x}), c(y)\right) \text{ for all } y \in X. \quad (4.1.34)$$

In the right handside above we use the notation $H(\cdot)$ for the (convex) exact improvement function (4.1.2), written with target $\tau = (f(\bar{x}) + (\bar{\rho} - \bar{\sigma})c(\bar{x}), 0)$. In particular, when $y = \bar{x}$ the inequality (4.1.34) becomes

$$0 < c(\bar{x}) \leq H(\bar{x}) = \max\left(-(\bar{\rho} - \bar{\sigma})c(\bar{x}), c(\bar{x})\right).$$

Since by (4.1.20) the penalties satisfy $\rho_k \geq RS + \sigma_k - 1 > \sigma_k - 1$, this means that $-(\bar{\rho} - \bar{\sigma}) < 1$ and the maximum above is attained at $c(\bar{x})$ and, hence, $H(\bar{x}) = c(\bar{x})$. As a result, (4.1.34) states that $0 \in \partial(H + i_X)(\bar{x})$, where i_X denotes the indicator function of the set X . Being a max-function, the subgradients of H at \bar{x} are of the form

$$\begin{aligned} \lambda g_{\mathbf{f}} + (1 - \lambda)g_{\mathbf{c}} \text{ for } \lambda \in [0, 1] \text{ with } g_{\mathbf{f}} \in \partial f(\bar{x}) \text{ and } g_{\mathbf{c}} \in \partial c(\bar{x}) \\ \text{with } \lambda \in (0, 1] \iff -(\bar{\rho} - \bar{\sigma})c(\bar{x}) \geq c(\bar{x}). \end{aligned}$$

Since $c(\bar{x}) > 0$ and $-(\bar{\rho} - \bar{\sigma}) \leq 1 - RS < 1$ by (4.1.20), the only possibility is $\lambda = 0$, i.e., $0 \in \partial c(\bar{x})$ and, therefore, $0 < c(\bar{x}) \leq c(y)$ for all $y \in X$, as stated. \square

For completeness, and to relate our method with previous algorithms in the literature, we finish with a result for oracles that make exact calculations. In this case, there is no noise attenuation step and the nominal decrease (4.1.13) is defined with $\alpha_k \in [0, 1]$.

Corollary 4.1.8 (Convergence for exact oracles). *Consider an exact oracle: for any $x^j \in X$ (4.1.3) returns $f^j = f(x^j)$, $c^j = c(x^j)$ and subgradients $g_{\mathbf{f}}^j \in \partial f(x^j)$ and $g_{\mathbf{c}}^j \in \partial c(x^j)$. Then Algorithm 4.1 with $\beta_k \equiv 0$ never executes Step 2 (noise attenuation).*

Furthermore, suppose $\alpha_k \in [0, 1]$, the penalties satisfy (4.1.20), the prox-parameter is updated according to (4.1.19a) and (4.1.19c), and the model satisfies (4.1.26). Then the statements in Corollary 4.1.7 apply.

Proof. By convexity, an exact oracle is of the lower type, so (4.1.31) holds with $\varepsilon = 0$. In particular, writing (4.1.4) and (4.1.5) with $y = \hat{x}^k$ we see that $h_{\hat{x}^k}^k \geq \mathcal{M}_k(\hat{x}^k)$. Since, by (4.1.6), $\mathcal{M}_k(\hat{x}^k) \geq \mathcal{M}_k(x^{k+1}) + \frac{1}{2}\mu_k\|x^{k+1} - \hat{x}^k\|^2$, the left handside in (4.1.12) is always nonnegative and, as long as $\beta_k \geq 0$, the algorithm will never consider noise too large (naturally so, since for exact oracles there is no noise to be detected).

The set X is bounded and the subdifferential mapping of a convex function is locally bounded, so (4.1.21) holds for exact oracles. Since in addition, (4.1.15) holds by taking for example $B = \frac{1}{2} = -b$, Corollary 4.1.7 completes the proof. \square

4.1.4.1 Relation with previous work

Corollary 4.1.8 covers methods based on improvement functions already considered in the literature for solving constrained non-smooth problems using exact or lower oracles. More precisely, conditions (4.1.20) are satisfied by at least the following three choices:

$$\rho_k = \sigma_k \equiv 0, \quad \text{or } \rho_k \equiv \bar{\rho} < +\infty \text{ and } \sigma_k \equiv 1, \quad \text{or } \bar{\rho} = +\infty \text{ and } \sigma_k \equiv 0,$$

corresponding to the improvement functions in [211], [2], and [129], respectively. The first two methods were developed for exact oracles ([2] deals with nonconvex functions). The approach suggested in [129] considers a setting corresponding to a lower oracle, and is addressed by Corollary 4.1.7.

To decide if the iterate gives a serious step, the three methods ([211, 2, 129]) use in Step 4 a criterion based on descent of the improvement function. Namely,

$$\text{if } h_{x^{k+1}}^k = \max(f^{k+1} - \tau_1^k, c^{k+1} - \tau_2^k) \leq h_{\hat{x}^k}^k - m\delta^k \quad \text{the iteration is declared serious} \quad (4.1.35)$$

(noting that the test (4.1.16) is also mentioned in [129] as a possibility). In view of the second item in Proposition 4.1.1, the criterion (4.1.16) is stronger for null steps. Depending on the problem, one criterion or the other might be preferable. By checking the proofs of Lemma 4.1.2 and 4.1.4, it is not difficult to see that the asymptotic results in Section 4.1.2 still hold with the alternative test.

It should also be mentioned that [129] uses an alternative stopping test, suitable for unbounded feasible sets (which is not the case in (4.1.1)). More precisely, instead of checking if $\delta^k \leq \text{TOL}$, Kiwiel in [129] uses the conditions

$$\max\left\{\|G^k + \nu^k\|, \mathbf{E}^k + \langle G^k + \nu^k, \hat{x}^k \rangle\right\} \leq \text{TOL}_{[129]} \text{ and } c_{\hat{x}^k} \leq 0. \quad (4.1.36)$$

For bounded feasible sets in (4.1.1), (4.1.36) is equivalent to the stopping test $\delta^k \leq \text{TOL}$. Indeed, following the definition of \mathbf{E}^k in (4.1.10) and δ^k in (4.1.13) with $\alpha_k = 0$, we see that $\mathbf{E}^k + \langle G^k + \nu^k, \hat{x}^k \rangle = \delta^k + \langle G^k + \nu^k, \hat{x}^{k+1} \rangle$. Using the fact that X is compact and in particular bounded together with $\|G^k + \nu^k\| \leq \text{TOL}_{[129]}$, the stopping criteria (4.1.36) implies that $\delta^k \leq \text{TOL}$ for certain modified tolerance.

Another stopping criterion is the one used in [68]:

$$\max\left\{\mu_k \|x^{k+1} - \hat{x}^k\|^2, \mathbf{E}^k\right\} \leq \text{TOL}_{[68]}. \quad (4.1.37)$$

It was already mentioned that Lemmas 4.1.2, 4.1.3, 4.1.4, ensure that $\|G^k + \nu^k\| = \mu_k \|x^{k+1} - \hat{x}^k\| \rightarrow 0$. Together with (4.1.17) the stopping test $\delta^k \leq \text{TOL}$ implies (4.1.37) for an appropriate tolerance $\text{TOL}_{[68]}$. On the other hand, keeping the prox-parameters uniformly bounded from above in (4.1.19), together with (4.1.17) and (4.1.7) gives

$$\delta^k = \mathbf{E}^k + \frac{2 - \alpha_k}{2} \mu_k \|x^{k+1} - \hat{x}^k\|^2 \leq 2 \max\left\{\mathbf{E}^k, \mu_k \|x^{k+1} - \hat{x}^k\|^2\right\}, \quad (4.1.38)$$

showing that (4.1.37) also implies $\delta^k \leq \text{TOL}$ for an appropriately chosen tolerance TOL .

Therefore, for convex problems as (4.1.1), our approach includes previous work, significantly extending the applicability of algorithms in the literature:

- not only exact and lower oracles can be used, but also upper ones, satisfying (4.1.31);
- the criterion for serious steps can be (4.1.16) or based on the improvement function;
- the descent and noise parameters α_k, β_k can be chosen in any manner satisfying (4.1.15).

This versatility has a positive impact on the numerical results as shown in Table 4.4.

– any choice for the penalty parameters ρ_k and σ_k in (4.1.4) satisfying (4.1.20) is possible.

Regarding the last item, the role/utility of the penalty in second target (in τ_2^k) is not clear. At least in the convex setting, taking $\sigma_k = 0$ and forgetting about this additional parameter seems sufficient. It is argued in [2], however, that a positive σ_k can be beneficial in the numerical performance of the algorithm, so the situation may be different for nonconvex problems.

4.1.5 Energy Application: Hydro Reservoir Management

We consider a cascaded reservoir management problem with uncertainty on inflows in the numerical experience. Figures 4.1(a) and 4.1(b) provide two typical instances of such a problem. Specific modelling details can be found in section 5.1 and paper [246].

Such a problem fits the following abstract structure:

$$\begin{aligned} \min_{x \geq 0, x \in \mathbb{R}^n} \quad & \langle f, x \rangle \\ \text{s.t.} \quad & \tilde{A}x \leq \tilde{b} \\ & p \leq \mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x], \end{aligned} \quad (4.1.39)$$

where $\xi \in \mathbb{R}^m$ is a Gaussian random vector with variance-covariance matrix Σ and zero mean (we have explicitly extracted the non-zero average in the vector $a^r, b^r \in \mathbb{R}^m$). This random vector represents uncertainty on inflows and the probabilistic constraint the fact that we are looking for a turbining schedule that keeps volume between a lower and upper bound with high probability.

By a classical result (Theorem 2.5.13), the feasible set induced by the joint chance constraint (5.1.4) is convex. Moreover from the same result it follows that the function

$$\log(p) - \log(\mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x]) \quad (4.1.40)$$

is a convex function.

So (4.1.39) corresponds to (4.1.1) with

$$f(x) := \langle f, x \rangle, \quad X := \{x \in \mathbb{R}^n : x \geq 0, \tilde{A}x \leq \tilde{b}\}$$

and

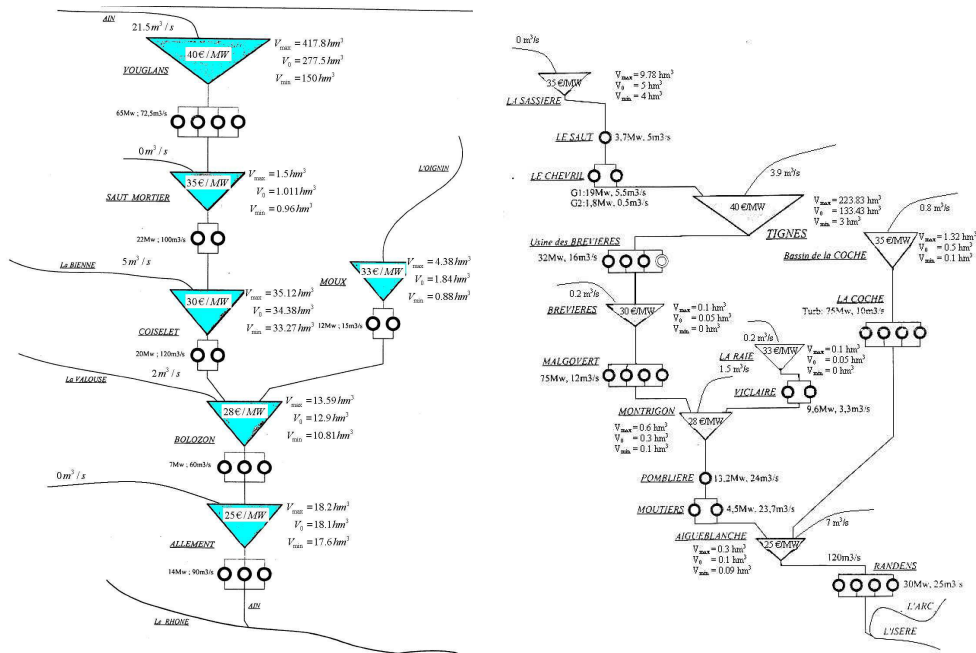
$$c(x) := \log(p) - \log(\mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x]).$$

It is moreover clear that X is bounded according to (5.1.1). In this setting, the f -oracle is exact. As for the c -oracle, it falls into the framework (4.1.31) with $\varepsilon > 0$, as explained below.

4.1.5.1 Devising an inexact upper oracle for the constraint

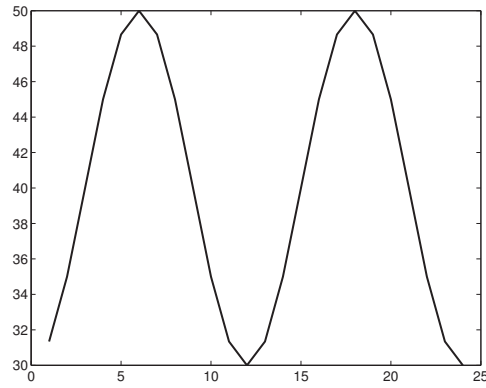
It is shown in Lemma 3.2.1 (see also Corollary 3.2.3) that the mapping

$$\varphi : \mathbb{R}^n \rightarrow [0, 1] \quad \text{defined by} \quad \varphi(x) = \mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x]$$



(a) The Ain Hydro Valley

(b) The Isère Hydro Valley



(c) The Price Signal

Figure 4.1: Numerical Instance Data

is differentiable with gradient $\nabla\varphi(x) = \nabla_{\hat{a}}F_{\xi}(\hat{a}, \hat{b})^{\top}A^r + \nabla_{\hat{b}}F_{\xi}(\hat{a}, \hat{b})^{\top}A^r$, with $\hat{a} = A^rx + a^r$, $\hat{b} = A^rx + b^r$ and $F_{\xi} : \mathbb{R}^m \times \mathbb{R}^m \rightarrow [0, 1]$ defined by $F_{\xi}(\hat{a}, \hat{b}) = \mathbb{P}[\hat{a} \leq \xi \leq \hat{b}]$. These results can be applied since ξ is a non-degenerate (i.e., with positive definite covariance matrix) Gaussian random variable. In the potentially degenerate case we can refer to [106]. The analysis of what follows will be completely similar as again the computation of the gradient is reduced to evaluation mappings with a structure close to the one of φ above.

Accordingly, the i -th component of $\nabla_{\hat{a}}F_{\xi}(\hat{a}, \hat{b})$ ($-\nabla_{\hat{b}}F_{\xi}(\hat{a}, \hat{b})$ respectively) is equal to $-\psi(z)\mathbb{P}[\tilde{a}^r + \tilde{A}^rx \leq \tilde{\xi} \leq \tilde{b}^r + \tilde{A}^rx]$. In this expression, ψ is the density of a standard Gaussian random variable in 1 dimension; z a specific point depending on x , and $\tilde{a}^r, \tilde{b}^r, \tilde{A}^r, \tilde{\xi}$ of a^r, b^r, A^r are appropriate modifications of the respective objects. These modifications depend on whether the derivative is taken with respect to \hat{a} or \hat{b} .

We see that, in order to compute one component of the gradient, one needs to evaluate a mapping of the form $F_{\zeta}(\hat{a}, \hat{b})$ at specific points $\hat{a} \leq \hat{b} \in \mathbb{R}^m$, with $\zeta \in \mathbb{R}^m$ a non-degenerate multi-variate Gaussian random variable. Since the evaluation of the c -function also requires to compute a similar probability, the core of the c -oracle is to make the involved multidimensional calculations in a fast and efficient manner. This is achieved by using the code [83], developed by A. Genz for multivariate normal probabilities. Having as input a requested accuracy $\varepsilon^g > 0$, the code either returns a value \tilde{F} such that $|\tilde{F} - F_{\zeta}(\hat{a}, \hat{b})| < \varepsilon^g$, or issues an error message, stating the impossibility of making the calculation with the requested precision. In the latter case, it is possible to increase the number of quasi Monte Carlo particles used in the numerical integration and make another attempt to obtain the desired accuracy.

Since the integral approximation estimate can be larger or smaller than the exact value, the c -linearizations may lie below or above the exact function c , noting that asymptotically exact (up to say floating point precision) calculations would be possible (although they might considerably increase the time spent in the oracle).

4.1.5.2 Convergence results for the application

According to the model set up in this section, the objective function f of problem (4.1.1) is linear and therefore has an exact oracle. Inexactness arises from evaluating the probabilistic constraint c . The corresponding oracle may be of the upper type, because no information is available on the sign of the incurred error, neither for the function values nor for the subgradients. However, as explained in [101], the user can control such error, keeping it sufficiently small if desired (provided enough CPU time is spent in the calculations).

We now derive an explicit expression for $\varepsilon > 0$ in (4.1.31). Consider a constant $\Phi > 0$, for which $\varphi(x^k) > \Phi > 0$ for all iterations k . For example, one can take as initial point \hat{x}^0 an appropriate convex combination with the Slater point $x^s \in \mathbb{R}^n$ ensuring that $\varphi(\hat{x}^0) > \Phi$ for any given $p > \Phi > 0$. Then, as far as serious iterates are concerned, (4.1.16) ensures that $\varphi(\hat{x}^k) > \Phi$. Since ε in (4.1.31) depends on the limiting behaviour, clearly one can choose μ_k in (4.1.6) according to rule (4.1.19c) in such a way that $\varphi(x^k) > \Phi$ also in

null-steps.

The logarithm being a uniformly continuous mapping on $[\Phi, 1]$, X being a compact set and $\varphi : X \rightarrow [0, 1]$ also being uniformly continuous, it is clear that for any $\varepsilon' > 0$ a precision ε^g can be chosen such that $|c(x^k) - c_{x^k}| \leq \varepsilon'$ for any iteration k . As discussed at the end of Section 4.1.5.1, $\|\nabla\varphi(x) - \nabla\varphi_{x^k}\|_\infty \leq \frac{2}{\sqrt{2\pi}} \|A^r\|_\infty \varepsilon^g$, since the standard Gaussian density ψ satisfies $|\psi(z)| \leq \frac{1}{\sqrt{2\pi}} \forall z \in \mathbb{R}$. By the Cauchy-Schwarz inequality and compactness of X we then obtain that

$$|\langle g_c(x^k), y - x^k \rangle - \langle g_{c_{x^k}}, y - x^k \rangle| \leq M\varepsilon^g$$

for an appropriate constant $M > 0$ and any $y \in X$. We conclude that (4.1.31) holds with $\varepsilon \geq \varepsilon' + M\varepsilon^g$. Moreover, whenever ε^g is taken small enough, ε' can also be made arbitrarily small.

We are in the conditions of Theorem 4.1.5. Since in the hydro valley application it is reasonable to assume the existence of a Slater point x^s , we pick ε^g , the user-defined precision of Genz' code in such a way that $c(x^s) \leq -2\varepsilon$. This ensures that the set X_ε in Theorem 4.1.5 is not empty. In the setting of the theorem, let \bar{x} be the limiting point generated by Algorithm 4.1.

When \bar{x} is infeasible according to the oracle information, $\bar{c} > 0$ and item (i) of Theorem 4.1.5 leads to the contradiction $\bar{c} < c(x^s) + \varepsilon \leq -\varepsilon < 0$. Therefore, \bar{x} is approximately feasible ($\bar{c} < 0$) and satisfies the relation $\bar{f} \leq f(y) + \varepsilon$ for all $y \in X_\varepsilon$ by item (ii) of the same theorem. Consequently, \bar{x} approximately solves (4.1.1).

4.1.6 Numerical experience

For our numerical experience, the starting point uses x^d , a solution to the linear program

$$\begin{aligned} \min_{x \geq 0, x \in \mathbb{R}^n} \quad & \langle f, x \rangle \\ \text{s.t.} \quad & \tilde{A}x \leq \tilde{b} \\ & a^r + A^r x \leq 0, b^r + A^r x \geq 0, \end{aligned} \tag{4.1.41}$$

the ‘‘deterministic’’ counterpart of (4.1.39) wherein the random vector ξ is replaced by its expectation. In general x^d will not be feasible for (4.1.39) (unless it solves the problem).

Since improvement functions are scale-dependent, [211], it is useful to scale the constraint. Accordingly, we consider the constraint $Kp \leq K\mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x]$ for some value of $K > 0$. A natural choice for K would be $|\langle f, x^d \rangle|$. Finally rules (4.1.19a), (4.1.19c) and (4.1.19b) are dealt with in the following way. In serious steps we take $\mu_{k+1} = \mu_k$ without making any changes. When noise is detected, we take $\mu_{k+1} = \kappa\mu_k$ for a parameter $\kappa \in (0, 1)$. Finally when null steps are made we choose $\mu_{k+1} = \min\{\mu_s\mu_k, \mu_{max}\}$ for a parameter $\mu_s > 1$.

4.1.6.1 The various compared algorithms

We first compare the performance of several methods, including variants of Algorithm 4.1 using different choices for the parameters. More precisely, we consider:

- A configuration of Alg.4.1 with a strong noise test:

$$\text{Alg.PB.SEV} : \quad \sigma_k \equiv 0, \quad \rho_k \equiv 0, \quad \alpha_k \equiv 1, \quad \beta_k \equiv -1 + \varepsilon^m$$

with stopping test (4.1.37) and descent test (4.1.35). Here ε^m is the machine precision.

- A configuration of Alg.4.1 with null parameters:

$$\text{Alg.PB.NUL} : \quad \sigma_k \equiv 0, \quad \rho_k \equiv 0, \quad \alpha_k \equiv 0, \quad \beta_k \equiv 0$$

with stopping test (4.1.36) and descent test (4.1.35)

- The method of centers from [129], bearing some similarities with Alg. 4.1 provided parameters are properly set:

$$\text{Alg.}[129] : \quad \sigma_k \equiv 0, \quad \rho_k \rightarrow \infty \text{ if } \{\hat{x}^{k+1}\} \text{ infeasible, } \alpha_k \equiv 0, \quad \beta_k = \text{ad-hoc}$$

with stopping test (4.1.36) and descent test (4.1.35)

- The conic bundle method from [132]:

$$\text{Alg.}[132] : \quad \text{No } \sigma_k, \rho_k, \quad \alpha_k \equiv 0, \quad \text{ad-hoc QP in Step 1, } \beta_k, \text{ Stop and Descent Test}$$

- The supporting hyperplanes method from [182] (see also [253, 185, 222]):

$$\text{Alg.}[182] : \quad \text{No } \sigma_k, \rho_k, \alpha_k, \beta_k,$$

with a Linear Program in Step 1 and ad-hoc stopping and descent tests.

In both configurations of Alg.4.1 above taking $RS = 1$ ensures satisfaction of (4.1.20) for any choice of ρ_{k+1} ; as for (4.1.15), it suffices to take any positive B and $b = \beta_k$ (constant in k). Since the conditions in Section 4.1.3 are satisfied, eventual convergence to an approximate solution is assured with these methods.

We now describe the specific ad-hoc updates of the last three methods. To help the comparison, the algorithms steps are numbered as in Alg. 4.1.

4.1.6.1.1 Alg.[129] Specifics The penalty ρ_{k+1} is halved at serious steps and doubled if $c_{\hat{x}^{k+1}} > 0$. Once a feasible stability center is found, the numerical value of this penalty becomes irrelevant. For an additional Armijo-like parameter $m' \in (0, 1]$, the method of centers modifies the noise management Step 2 in Alg. 4.1 as follows.

Step 2'a (Infeasible serious point) If $\hat{c}^k > 0$ and $\delta^k < m'\hat{c}^k$, noise is too large. Decrease the prox-parameter as in (4.1.19b), take $\rho_{k+1} = 2\rho_k$, maintain the bundle, and loop to Step 1 (solve a new QP subproblem).

Step 2'b (Noise attenuation). In the other cases, that is, if $\hat{c}^k \leq 0$ or $\delta^k \geq m'\hat{c}^k$, check if condition (4.1.14) holds. If this condition holds, noise is too large. Decrease the prox-parameter as in (4.1.19b), take $\rho_{k+1} > \rho_k$ if $\hat{c}^k > 0$, maintain the bundle, and loop to Step 1 (solve a new QP subproblem).

The difference with our noise management step is in the first item: when the current serious point is infeasible, instead of (4.1.14) the condition $\delta^k < m'\hat{c}^k$ is checked. Such condition amounts to verifying if (4.1.14) holds with $\beta_k = 1 - \frac{2m'\hat{c}^k}{\mu_k \|x^{k+1} - \hat{x}^k\|^2}$. Suppose such value of β_k also satisfies (4.1.15) (for example suppose m' is sufficiently small). Then this alternative can also be considered a special case of Algorithm 4.1, with a specific updating of β_k for infeasible serious points.

4.1.6.1.2 Alg.[132] Specifics This method is applicable when, like in our application, the objective function in (4.1.1) is linear or quadratic: the QP subproblem approximates (4.1.1) with a cutting-plane model for the constraint. Furthermore, the method needs the knowledge of a Slater point x^s for (4.1.1), which is also its starting point: $\hat{x}^0 = x^s$. The algorithm performs the following steps.

Step 1' (Alternative subproblem, δ^k, \mathbf{E}^k) Let (x^{k+1}, η^{k+1}) be a primal-dual solution to the QP

$$\begin{aligned} \min_{y \in X \subseteq \mathbb{R}^n} \quad & \langle f, y \rangle + \frac{1}{2} \mu_k \|y - \hat{x}^k\|^2 \\ \text{s.t.} \quad & \check{c}_k(y) \leq 0. \end{aligned}$$

This amounts to taking in (4.1.7), instead of subgradient $G^k \in \partial \mathcal{M}^k(x^{k+1})$, a vector $G^k \in f + \eta^{k+1} \partial \check{c}_k(x^{k+1})$ given by the QP optimality conditions. Taking $\delta^k := \langle f, \hat{x}^k - x^{k+1} \rangle$ and $\mathbf{E}^k = \delta^k - \mu_k \|x^{k+1} - \hat{x}^k\|^2$ ensures satisfaction of the left handside identity in (4.1.17) with $\alpha_k = 0$.

Step 3'a (Stopping test) Stop if $\|G^k + \nu^k\| \leq \text{tol}$ and $\mathbf{E}^k + \langle G^k + \nu^k, \hat{x}^k \rangle \leq 0$,

Step 3'b (Interpolation Step) If $c_{x^{k+1}} \leq 0$, set $\gamma^k := 1$, otherwise $\gamma^k := \frac{-c_{x^s}}{c_{x^{k+1}} - c_{x^s}}$.

Define $\tilde{x}^k := \gamma^k x^{k+1} + (1 - \gamma^k) x^s$.

Step 4' (Serious step Test) If $\langle f, \tilde{x}^k \rangle \langle f, \hat{x}^k \rangle - m\delta^k$ then declare a serious step, taking as next center $\hat{x}^{k+1} = \tilde{x}^k$. Otherwise, declare a null step.

When the c -oracle is exact or lower², each stability center will be feasible, by convexity. When the c -oracle is of the upper type, like in our case, by the discussion in Section 4.1.5.1 it is possible to take a slightly larger γ^k so that $\varphi(\tilde{x}^k) \geq p + \varepsilon^g$. This is a potentially costly fix that may require several evaluations (our implementation uses the original definition for γ^k .) The analysis in [132] does not cover upper oracles, so convergence of this method is unclear, although its numerical behaviour was reasonable for our tests.

4.1.6.1.3 Alg.[182] Specifics Like Alg.[132], the cutting-planes model for the constraint is built using for the c -evaluation points, points obtained from interpolating with

²In this case one requires the c -oracle to be exactly evaluated at the Slater point

the Slater point. The difference is that now the constraint is required to be active at the interpolation point ($x_c^0 = (1 - \gamma^0)x^0 + \gamma^0x^s$ satisfies $c_{x_c^0} = 0$ for some $\gamma^0 \in [0, 1]$.) For a linear objective function the algorithm performs the following steps.

Step 1' (Linear Programming subproblem) Let x^{k+1} be a solution to

$$\begin{aligned} \min_{y \in X \subseteq \mathbb{R}^n} \quad & \langle f, y \rangle \\ \text{s.t.} \quad & \check{c}_k(y) \leq 0. \end{aligned}$$

Step 3'a (Interpolation and Oracle) Determine $\gamma^k \in [0, 1]$ for which $x_c^{k+1} = (1 - \gamma^k)x^{k+1} + \gamma^kx^s$ satisfies $c_{x_c^{k+1}} = 0$. Call the c -oracle and add the linearization to the cutting-plane model.

Step 3'b (Stopping Test and Loop) If $\langle f, x_c^{k+1} - x^{k+1} \rangle < \langle f, x^{k+1} \rangle_{\text{TOL}}$ then stop. Otherwise, set $k = k + 1$ and loop to Step 1'.

For exact c -oracles, the method is a specialization of the cutting-plane algorithm in non-smooth optimization, hence it converges. For inexact c -oracles, convergence results are not known.

4.1.6.2 Computational results

We now comment some of the numerical results. The reported CPU times are to be taken as a measure for comparing different algorithms, rather than as a measure of performance. Our implementation does not contain several improvements and tweaks that could optimize the code (oracle parallelization or multi-threading, for instance). Another important issue is that computing a Slater point is very consuming in terms of CPU time: this amounts to solving a problem as difficult as the original one (4.1.39) (i.e., solving (4.1.39) with objective function replaced by maximizing the probability level).

Results reported in Tables 4.1,4.2,4.3 were obtained on a HP xw6200 workstation with 8 Gb memory, whereas results of Tables 4.4, 4.6, 4.5 were obtained on a HP z600 workstation.

The key findings are that Alg.PB.SEV

1. Does not need the explicit knowledge of a Slater point. Moreover the Bundle Algorithms perform better if started in an infeasible point. Such an infeasible start can not be performed with Alg.[182] and Alg.[132] which are definitely outperformed.
2. Provides the optimal solution and does not converge early as do Alg.[129] and Alg.PB.NUL.
3. Is quite robust to the accuracy with which the probabilistic constraints and gradients are evaluated.

4.1.6.3 Feasible Start using a Slater Point and Infeasible Start

In order to put all algorithms on equal foot, a first comparison supposes that a Slater point is available to all of the methods in the benchmark. In this case, for a suitable convex combination of x^d and x^s , the starting point \hat{x}^0 is always feasible (Alg.[132] starts at x^s). Computing this convex multiplier requires executing step 3'a of Alg.[182]. This involves calling the c -oracle several times and might be costly. Table 4.1 contains the output of all the runs, with various parameter settings for each algorithm (to decide on the best choice of parameters for the benchmark).

The best objective function value in this test was -104162 , found by Alg.PB.SEV for different settings. When $K = 1e^4$ Alg.PB.SEV found in 239 minutes a similar³ solution than Alg.[182], which took approximately the same CPU time. Taking $K = 5e^4$ the same point is reached in only 193 minutes and with $K = 1e^4$ that point is reached in 178 minutes. Table 4.1 shows the importance of the scaling parameter in the bundle method. It also shows that the supporting hyperplane method reaches good points early, but takes a very long time to converge. Basically each iteration takes approximately 15 to 20 minutes. Each bundle iteration takes less time (approximately 3 to 4 minutes per iteration). This can be explained by the fact that the supporting hyperplane method computes the exact interpolation. Now typically the cutting-plane method would yield a new iterate x^k far from the probability boundary $\mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x] = p$ and it would seem that Genz' code takes more time on such points than on strictly feasible points. It is also interesting to note that each stability center from the bundle methods is feasible. Thereby empirically providing support for the initial discussion in Section 4.1.5.2. Since in this setting Alg.[129] and Alg.PB.NUL would have similar results, we did not include them in this comparison.

The results in Table 4.2 were obtained with an infeasible starting point, $\hat{x}^0 = x^d$ from (4.1.41), using for each algorithm the best parameter settings in Table 4.1.

For these runs, both Alg.PB.NUL and Alg.[129] modify the serious step test in Step 4, as follows. When the current stability center is infeasible, in addition to the usual test for acceptance, any improvement in feasibility by at least the c -oracle precision (“Genz precision” ε^g) declares the current iterate a new stability center. The modification was done to prevent the methods from stalling in the early stages of the algorithmic process and can only be active a finite number of iterations. Alg.PB.SEV does not stall and needs no modification. Once more, this variant is the best one in the benchmark, since it finds a feasible point with objective function value -104153 in only 56 iterations (comparable to the other two algorithms in the comparison). The best objective value of -104162 was obtained at the stake of many iterations. This table shows that important speed ups can be gained when starting with infeasible points. In particular, Alg.[182] requiring a Slater point and reaching a feasible solution with objective function value -104154 in 2079.17 minutes (see Table 4.1) is definitely out-performed by the Bundle method settings of Table 4.2.

The comparison above shows a strong dependence of the CPU time on the initial point,

³i.e., feasible with same objective function value

Table 4.1: Comparison of Algorithms (nne^x stands for $nn10^x$), assuming a Slater Point available.

method	Obj. Value	Nb. Iter.	CPU time (mins)	parameters
Alg.[182]	-103197	17	247.7	$tol = 1e^{-2}$
Alg.[182]	-104070	45	940.3	$tol = 1e^{-3}$
Alg.[182]	-104154	94	2079.17	$tol = 1e^{-4}$
Alg.PB.SEV	-104162	294	1028.43	$K = 1e^5, \mu_0 = 1e^{-6}, \mu_s = 2, \kappa = 0.7$
Alg.PB.SEV	-104160	300	1060.35	$K = 2e^5, \mu_0 = 1e^{-6}, \mu_s = 2, \kappa = 0.7$
Alg.PB.SEV	-104162	286	991.33	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 1.05, \kappa = 0.95$
Alg.PB.SEV	-104162	243	846.20	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2, \kappa = 0.7$
Alg.PB.SEV	-104160	215	723.55	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2, \kappa = 0.25$
Alg.PB.SEV	-104162	255	887.07	$K = 5e^4, \mu_0 = 1e^{-5}, \mu_s = 2, \kappa = 0.1$
Alg.PB.SEV	-104162	257	907.20	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2, \kappa = 0.1$
Alg.PB.SEV	-104162	239	827.26	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 4, \kappa = 0.5$
Alg.PB.SEV	-104162	265	932.19	$K = 1e^4, \mu_0 = 1e^{-6}, \mu_s = 2, \kappa = 0.7$
Alg.PB.NUL	-104109	417	1686.12	$K = 1e^5, \mu_0 = 1e^{-6}, \mu_s = 1.05, \kappa = 0.95$
Alg.PB.NUL	-104096	402	1539.16	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 1.05, \kappa = 0.95$
Alg.PB.NUL	-104102	395	1639.40	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2, \kappa = 0.7$
Alg.PB.NUL	-104089	277	1106.43	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.25$
Alg.PB.NUL	-104091	323	1257.09	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 4.0, \kappa = 0.5$
Alg.PB.NUL	-104078	366	1480.21	$K = 1e^4, \mu_0 = 1e^{-6}, \mu_s = 1.05, \kappa = 0.95$
Alg.PB.NUL	-104077	395	1591.13	$K = 1e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.7$
Alg.PB.NUL	-104078	278	1063.49	$K = 1e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.5$
Alg.PB.NUL	-104076	234	887.38	$K = 1e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.25$
Alg.PB.NUL	-104078	253	981.13	$K = 1e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.35$
Alg.PB.NUL	-104079	144	549.19	$K = 1e^4, \mu_0 = 1e^{-5}, \mu_s = 2.0, \kappa = 0.1$
Alg.PB.NUL	-104072	208	792.17	$K = 1e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.1$
Alg.PB.NUL	-104078	218	828.58	$K = 1e^4, \mu_0 = 1e^{-7}, \mu_s = 2.0, \kappa = 0.1$
Alg.PB.NUL	-104078	278	1126.51	$K = 1e^4, \mu_0 = 1e^{-6}, \mu_s = 4.0, \kappa = 0.5$
Alg.[132]	-104026	525	1933.29	$K = 1, \mu_0 = 1e^{-5}, \mu_s = 1.05, \kappa = 0.7$
Alg.[132]	-104026	481	1691.07	$K = 1, \mu_0 = 1e^{-6}, \mu_s = 1.05, \kappa = 0.7$
Alg.[132]	-104027	447	1749.57	$K = 1, \mu_0 = 1e^{-7}, \mu_s = 1.05, \kappa = 0.7$
Alg.[132]	-104027	454	1852.24	$K = 1, \mu_0 = 1e^{-8}, \mu_s = 1.05, \kappa = 0.7$
Alg.[132]	-104025	339	1311.24	$K = 1, \mu_0 = 1e^{-8}, \mu_s = 1.05, \kappa = 0.1$
Alg.[132]	-104024	297	1170.13	$K = 1, \mu_0 = 5e^{-9}, \mu_s = 1.05, \kappa = 0.1$
Alg.[132]	-104027	418	1602.25	$K = 1, \mu_0 = 1e^{-9}, \mu_s = 1.05, \kappa = 0.7$
Alg.[132]	-104022	293	1089.25	$K = 1, \mu_0 = 1e^{-9}, \mu_s = 1.05, \kappa = 0.1$
Alg.[132]	-104026	273	1030.21	$K = 1, \mu_0 = 1e^{-9}, \mu_s = 1.05, \kappa = 0.05$
Alg.[132]	-104024	254	893.40	$K = 1, \mu_0 = 1e^{-9}, \mu_s = 1.01, \kappa = 0.05$
Alg.[132]	-104024	254	798.32	$K = 1, \mu_0 = 1e^{-9}, \mu_s = 1.01, \kappa = 0.01$
Alg.[132]	-104024	241	761.37	$K = 1, \mu_0 = 1e^{-9}, \mu_s = 1.001, \kappa = 0.01$
Alg.[132]	-104026	303	1150.23	$K = 1, \mu_0 = 1e^{-9}, \mu_s = 4.0, \kappa = 0.1$
Alg.[132]	-104025	391	1456.13	$K = 1, \mu_0 = 1e^{-10}, \mu_s = 1.05, \kappa = 0.7$
Alg.[132]	-104026	329	1193.51	$K = 1, \mu_0 = 1e^{-10}, \mu_s = 1.05, \kappa = 0.1$
Alg.[132]	-104028	595	2340.03	$K = 1, \mu_0 = 1e^{-10}, \mu_s = 2.0, \kappa = 0.7$
Alg.[132]	-104020	360	1348.14	$K = 1, \mu_0 = 1e^{-10}, \mu_s = 2.0, \kappa = 0.1$
Alg.[132]	-104026	325	1250.17	$K = 1, \mu_0 = 1e^{-10}, \mu_s = 4.0, \kappa = 0.1$

for all methods.

Table 4.2: Comparison of Algorithms (nne^x stands for $nn10^x$), infeasible starting point. Precision of Genz' code $\varepsilon^g = 1e^{-4}$.

method	Obj. Value	Nb. Iter.	CPU time (mins)	parameters
Alg.PB.SEV	-104162	133	379.45	$K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.25$
Alg.PB.NUL	-104154	73	167.25	$K = 1e^4, \mu_0 = 1e^{-5}, \mu_s = 2.0, \kappa = 0.1$
Alg.[129]	-104153	55	114.52	$K = 1e^4, \mu_0 = 1e^{-5}, \mu_s = 2.0, \kappa = 0.1$

4.1.6.4 Different variants of Alg.PB.SEV

In order to determine the impact on the convergence, we varied the precision of Genz code. This test also allows us to check the potential of varying this precision along the iterations. We took the best method, Alg.PB.SEV, with the same parameter setting, for different oracle precision. Table 4.3 reports the corresponding results.

Table 4.3: Effect of ‘‘Genz Precision’’ (nne^x stands for $nn10^x$), Alg.PB.SEV with $K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.5$.

Starting Point	Genz Precision	objective Value	Nb. Iterations	CPU time (mins)
Feasible	$1e^{-2}$	-104177	216	83.55
Feasible	$1e^{-3}$	-104163	216	87.13
Feasible	$1e^{-4}$	-104162	222	722.59
Feasible	$1e^{-5}$	-104161	261	54390.5
Infeasible	$1e^{-2}$	-104177	71	27.48
Infeasible	$1e^{-3}$	-104163	108	41.36
Infeasible	$1e^{-4}$	-104162	128	338.37

The solution obtained when using Genz' code with precision $1e^{-2}$ is slightly infeasible. This explains the ‘‘over’’-optimal objective function value. From this table, we see that if the oracle accuracy is too high, CPU times can reach inacceptably large values.

Continuing with our analysis of the best variant, Alg.PB.SEV with fixed settings, in Table 4.4 we consider different values of β_k , ranging between the severe noise test (close to -1) to the permissive one (close to 1), and likewise for α_k , which varies from a severe serious step test (close to 0) to a permissive one (close to 2).

Table 4.4: Effect of noise test (nne^x stands for $nn10^x$), Alg.PB.SEV with $K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0, \kappa = 0.5, \text{TOL} = 0.5$, and Genz precision $5e^{-4}$

Starting Point	α	β	objective Value	Nb. Iterations	CPU time (mins)
Infeasible	0	$-1 + \varepsilon^m$	-104160	88	21.33
Infeasible	0	0	-104157	87	21.0
Infeasible	0	$1 - \varepsilon^m$	-104159	111	30.30
Infeasible	1	$-1 + \varepsilon^m$	-104158	60	12.8
Infeasible	1	$-\varepsilon^m$	-104158	70	15.57
Infeasible	$2 - 2\varepsilon^m$	$-1 + \varepsilon^m$	-104077	24	5.35

Higher α values lead to more serious steps, until the extreme of declaring all iterates serious ($\alpha = 2 - 2\varepsilon^m$). In our tests, a severe noise test has resulted in a more stable

management of the prox-parameter. Indeed, we have observed that if noise gets detected early on, the value of μ_k remains unchanged a significant number of iterations. By contrast, the permissive choice for β_k leads to some chaotic changes throughout the iterative process.

We have also tested the two alternative stopping criteria and the alternative serious step conditions. Table 4.5 reports these results.

Table 4.5: Effect of Descent test & Stopping Criteria (αe^x stands for $\alpha 10^x$). Alg.PB.SEV with parameter setting $K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0\kappa = 0.5$. Precision of Genz code is taken to be $5e^{-4}, \rho^k = 0, \alpha = 0, \beta = -1 + \varepsilon^m$.

Starting Point	Descent Test	Stopping Criteria	objective Value	Nb. It	CPU time (mins)
Infeasible	(4.1.16)	$\delta^k \leq 0.5$	-104160	93	23.33
Infeasible	(4.1.35)	$\delta^k \leq 0.5$	-104160	93	22.49
Infeasible	(4.1.16)	(4.1.36), TOL = $1e^{-4}$	-104153	69	16.38
Infeasible	(4.1.16)	(4.1.37), TOL = $1e^{-4}$	-104156	70	17.39

One can observe that in our specific case the choice of Descent Test (4.1.16) or (4.1.35) makes no difference. This can be intuited since for a currently feasible stability center with feasible candidate the check (4.1.35) is indeed identical to that of (4.1.16). Since this situation seems to occur most often in this application, the result of Table 4.5 are indeed natural. It could however be argued that descent test (4.1.16) is less sensitive to the scaling parameter K . In particular one could make an alternative choice for the Armijo parameter to boost feasibility steps early on. The stopping criteria clearly does impact results, mainly because the varying tolerances, although equivalent choices exist, are hard to link together.

Finally, we also examined the impact of penalty parameter ρ_k , comparing taking $\rho_k = 0$ (as in the previous runs in this subsection) with an update as for Alg.[129]⁴ Results are provided in Table 4.6. Instead, we observe that with a nonnull ρ_k a feasible stability center is found in half the number of iterations. Unfortunately this quest for feasibility strongly deteriorates the objective function value, and eventually more iterations are required to converge to very similar solutions in the end.

We conclude that, at least for our runs, the best variant is Alg.PB.SEV with the settings in Table 4.4.

4.1.7 Solution Quality

To assess the obtained solutions, we simulate the reservoirs evolution and check if the stipulated probability level is satisfied numerically. Figure 4.2 shows the reservoir levels for 100 simulated inflow scenarios, using the expected-value strategy obtained from

⁴That is: we multiply ρ^k by two at each iteration when the current stability center is infeasible. We take $\rho^k = 0$ whenever the current stability center is feasible. Since all following stability centers remain feasible from that moment onward there is no further effect of ρ^k .

Table 4.6: Effect of updating ρ^k and the noise test (αe^x stands for $\alpha 10^x$). Alg.PB.SEV with parameter setting $K = 5e^4, \mu_0 = 1e^{-6}, \mu_s = 2.0\kappa = 0.5, \text{TOL} = 0.5$. Precision of Genz code is taken to be $5e^{-4}$

Starting Point	α	β	objective Value	Nb. Iterations	CPU time (mins)
Infeasible	0	$-1 + \varepsilon^m$	-104160	93	23.33
Infeasible	0	0	-104160	131	31.3
Infeasible	0	$1 - \varepsilon^m$	-104157	115	32.40
Infeasible	1	$-1 + \varepsilon^m$	-104158	72	16.31
Infeasible	1	$-\varepsilon^m$	-104159	68	15.57
Infeasible	$2 - 2\varepsilon^m$	$-1 + \varepsilon^m$	-104095	27	6.57

solving problem (4.1.41) and Alg.PB.SEV. The figure clearly shows the importance of integrating uncertainty in order to obtain robust turbining strategies.

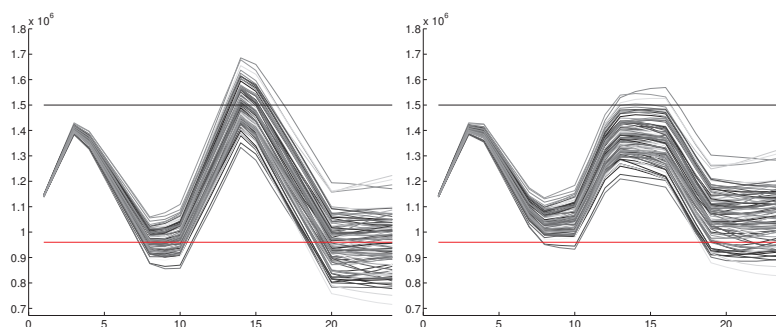


Figure 4.2: Evolution of reservoir “Saut Mortier”, for the expected-value strategy (left) and for Alg.PB.SEV strategy (right).

Except for the deterministic solution, which violates constraints for almost all scenarios, the various methods in our benchmark provide in fact very similar solutions. When slight differences arise, they account for increased robustness (with respect to the deterministic solution) and for increased optimality (with respect to various solution methods).

4.2 A level Bundle Method

Following the discussion of [47] at least three types of bundle methods can be identified. The classical proximal bundle method ([139] and references therein) of which a special version for probabilistically constrained programming was derived in Section 4.1. The level bundle method ([136]) and the trust-region bundle method ([214]). In the last version the cutting planes model of the convex function (to be minimized) is minimized under a trust-region constraint as its name implies. In the level method we orthogonally project the current stability center onto a level set defined by the current cutting planes model. For a given iteration, we can theoretically find parameter settings such that each of these methods provides the same next iterate. From a practical viewpoint these three methods differ in the ease with which the parameters are updated. For instance, the level bundle method does not have a proximal parameter to update. Since this is one of the most tricky parts for obtaining an efficient proximal bundle method, this can be seen as an advantage. In order to investigate this, we will derive a level bundle method for probabilistically constrained programming in this section. We will repeat some of the earlier notation for clarity.

Once again, let us consider problems of the type

$$\min_x f_0(x) \text{ s.t. } x \in X \subset \mathbb{R}^n \text{ and } f_j(x) \leq 0 \text{ for } j = 1, \dots, m, \quad (4.2.1)$$

where $f_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 0, \dots, m$ are convex functions and $X \neq \emptyset$ a polyhedral bounded set in \mathbb{R}^n . We will make the assumption that some (or all) functions f_j are hard to evaluate. We therefore assume that the mappings f_j , $j = 1, \dots, m$ can be inexactly evaluated by some oracle (or black-box) providing approximate values and subgradients, hereafter denoted by oracle information. The precision of the oracle information might be unknown, but it is assumed bounded.

Applications in which a decision maker has to make a decision hedging risks often lead to optimization problems with nonlinear convex constraints that are difficult to deal with. For instance, if such constraints (f_j , with $j = 1, \dots, m$) are setup as joint chance constraints (see Example 4.2.2 below). Evaluating the functions for a given point involves computing numerically a multidimensional integral. This is a difficult task when uncertainty variables are high-dimensional (say dimension 100), see [248]. A second difficulty is obtaining an efficient estimate of a subgradient. This can be done in many cases of interest, but equally involves imprecisions (see [244] for a special case and [233, 235, 236] for very general results).

The last few years have seen the occurrence of a new generation of bundle methods, capable to handle inexact oracle information. For unconstrained (or polyhedral constrained) convex non-smooth optimization problems we refer to [113] and [128] for general approaches, [132] for a combinatorial context, and [70], [72], [46] and [44] for a stochastic optimization framework. For an encompassing approach we refer to the recent work [45].

For constrained convex optimization problems like (4.2.1) more sophisticated methods need to come into play. We refer to [70] and [71] for level bundle methods capable to

deal with asymptotically exact oracle information. Such oracle information is assumed to underestimate the exact value of the functions, i.e., the oracle is of the lower type. Since [70] and [71] are based on the pioneering work [136], they are not readily implementable because they might require unbounded storage, i.e., all the linearizations built along iterations must be kept. The work presented in section 4.1 overcomes this drawback and furthermore allows for upper oracles, i.e., oracles that provide inexact information which might overestimate the exact values of the functions. In contrast to [70] and [71] that employ level bundle methods, that work applies a proximal bundle method, see also [114]. In all these approaches we define different improvement functions (see (4.2.8) below) to measure the improvement of each iterate towards a solution of the problem. Level bundle methods have at disposal lower bounds (self-built) to define improvement functions, in contrast to proximal bundle that must find other alternatives to the (in-available) lower bounds. We refer to [248], [211] and [120] for some improvement function alternatives.

Due to the improvement function definition, level bundle methods are likely to be preferable to proximal bundle methods to handle (nonlinear) constrained convex optimization problems. For this reason we follow some features of the restricted memory level bundle method proposed in [127] and extend [71] to handle upper oracles and to keep the storage information bounded; this is what we call restricted memory. Moreover, we extend [248] by considering upper oracles with on-demand accuracy (see the formal definition in Section 4.2.3.2), and like [71] we prove convergence for lower oracles with on-demand accuracy without assuming a Slater point.

This section is organized as follows: Section 4.2.1 presents two important examples of optimization problems that cast into the problem formulation (4.2.1), with difficult to evaluate objective and/or constraint functions. In Section 4.2.2 we propose our algorithm and prove its convergence for exact oracle information. The convergence analysis therein is the key for proving convergence of the algorithm when inexact oracle information come into play. Section 4.2.3 deals with inexact oracle information. We consider upper, lower, and on-demand oracles, depending on the assumptions made. In Section 4.2.4 we provide numerical experiments on realistic joint chance constrained energy problems, arising when dealing with robust cascaded-reservoir management. These are problems from Électricité de France. We compare the proposed approaches with some algorithms considered in [248].

To easy notation, in what follows we consider a short formulation to problem (4.2.1). Without loss of generality we may assume that $m = 1$ in (4.2.1) by defining the (non-smooth and convex) function $c : \mathbb{R}^n \rightarrow \mathbb{R}$

$$c(x) := \max_{j=1,\dots,m} f_j(x),$$

and call f_0 just f . With this notation, the problem we are interested in solving is (4.1.1):

$$f_{\min} := \min_{x \in X \subset \mathbb{R}^n} f(x) \quad \text{s.t.} \quad c(x) \leq 0, \quad (4.2.2)$$

which is assumed to have a solution; therefore, its optimal value f_{\min} is finite.

4.2.1 Some examples coming from stochastic optimization

We now provide two examples coming from stochastic optimization that can be cast into the above scheme, wherein f and/or c are difficult to evaluate.

Example 4.2.1 (Risk-averse two-stage stochastic programming). *Let $Z \subset \mathbb{R}^n$ be a bounded polyhedral set. Let also $\Xi \subset \mathbb{R}^d$ be a finite set containing elements (scenarios) ξ_i , $i = 1, \dots, N$; and suppose that $\varphi : Z \rightarrow \mathbb{R}$ is a convex function. A two-stage stochastic problem with mean-CV@R aversion model and finitely many scenario can be written as*

$$\begin{cases} \min & \varphi(z) + \sum_{i=1}^N p_i Q(z, \xi_i) \\ \text{s.t.} & t + \frac{1}{\beta} \sum_{i=1}^N p_i [Q(z, \xi_i) - t]_+ \leq \rho \\ & z \in Z, t \in \mathbb{R}, \end{cases} \quad (4.2.3)$$

where $1 \geq \beta > 0$ and $\rho \geq 0$ are given parameters, $p_i > 0$ is the probability associate to the scenario ξ_i , $[\cdot]_+$ denotes the positive part of a real number, and $Q : Z \times \Xi \rightarrow \mathbb{R}$ is a convex function on Z defined for instance by the linear problem

$$Q(z, \xi) := \begin{cases} \min & \langle q, y \rangle \\ \text{s.t.} & W(\xi)y + T(\xi)z = h(\xi) \\ & y \in \mathbb{R}_+^{n_2}. \end{cases}$$

The components q, W, T and h are assumed to be measurable mappings with appropriate dimensions and well defined in Ξ .

Problem (4.2.3) is a risk-averse two-stage stochastic programming problem. The nonlinear and difficult to evaluate constraint $t + \frac{1}{\beta} \sum_{i=1}^N p_i [Q(z, \xi_i) - t]_+ \leq \rho$ is the so called conditional value-at-risk (CV@R $_{\beta}$) with confidence level β ; see [198], [199], [71].

By bounding the variable t , writing $x = (z, t) \in X \subset Y \times \mathbb{R}$, and by taking

$$f(x) = f(z, t) = \varphi(z) + \sum_{i=1}^N p_i Q(z, \xi_i) \quad \text{and} \quad c(x) = c(z, t) = t + \frac{1}{\beta} \sum_{i=1}^N p_i [Q(z, \xi_i) - t]_+ - \rho$$

problem (4.2.3) corresponds to (4.2.2). Notice that computing the value of the functions $f(x)$ and $c(x)$ for each given point $x \in X$ amounts to solving N linear problems $Q(x, \xi_i)$, $i = 1, \dots, N$. After having solved those linear problems, subgradients for f and c are easily at hand; see [216] for more details. Therefore, for large N (say $N > 10000$) the functions f and c are hard to evaluate, and by making use of inexact oracle information (e.g., solving inexactly or just skipping some linear problems) problem (4.2.3) can be numerically tractable even for a large N .

These problems are taken into account in [71], by considering lower oracles with on-demand accuracy (see Section 4.2.3.4 for the formal definition). Besides the conditional value-at-risk approach of two-stage stochastic programs, [71] considers more risk-aversion concepts such as stochastic dominance, proposed in [52]. In all cases, no upper oracles are studied.

Example 4.2.2 (Chance-constrained programming). *Given a probability level p , the optimization problem*

$$\begin{cases} \min & \langle q, x \rangle \\ \text{s.t.} & \mathbb{P}[Ax + a \leq \xi \leq Ax + b] \geq p \\ & x \in X \end{cases} \quad (4.2.4)$$

has a feasible set that is convex for d -dimensional random variables ξ with log-concave probability distribution (but in many other settings too). In this formulation, A is a fixed matrix and a, b are fixed vectors with appropriate dimensions. Evaluating the constraint and computing a gradient with high precision is possible but potentially very costly, especially when the dimension d is large (say $d \geq 100$).

By a classical result (Theorem 2.5.13), not only the feasible set induced by the joint chance constraint in (4.2.4) is convex but also the function

$$\log(p) - \log(\mathbb{P}[Ax + a \leq \xi \leq Ax + b]). \quad (4.2.5)$$

Therefore, problem (4.2.4) corresponds to (4.2.2) by taking

$$f(x) = \langle q, x \rangle \quad \text{and} \quad c(x) = \log(p) - \log(\mathbb{P}[Ax + a \leq \xi \leq Ax + b]).$$

The oracle for the function $c(x)$ employs multidimensional numerical integration and quasi-Monte Carlo techniques for which the error sign is unknown (but a bound is computable).

Take $x \in X$. Since the inexact oracle information can overestimate the exact value $c(x)$ (upper oracle), the constrained level bundle methods proposed in [71] is not suitable.

In what follows we propose a new algorithm to deal with (in-)exact oracle information.

4.2.2 Level Bundle Method for Exact Oracles

In this section we assume that both functions f and c in (4.2.2) can be exactly evaluated by an oracle that provides for each $x^k \in X$:

$$\begin{cases} f\text{-oracle information:} & f(x^k) \quad \text{and} \quad g_{\mathbf{f}}^k \in \partial f(x^k) \\ c\text{-oracle information:} & c(x^k) \quad \text{and} \quad g_{\mathbf{c}}^k \in \partial c(x^k). \end{cases} \quad (4.2.6)$$

With such information from the oracle, we consider the approximate linearizations

$$\begin{aligned} \bar{f}^k(x) &:= f(x^k) + \langle g_{\mathbf{f}}^k, x - x^k \rangle \\ \bar{c}^k(x) &:= c(x^k) + \langle g_{\mathbf{c}}^k, x - x^k \rangle. \end{aligned}$$

At iteration k , two polyhedral cutting-plane models are available:

$$\check{f}^k(x) := \max_{j \in J_{\mathbf{f}}^k} \bar{f}^j(x) \quad \text{with} \quad J_{\mathbf{f}}^k \subset \{1, 2, \dots, k\} \quad (4.2.7a)$$

$$\check{c}^k(x) := \max_{j \in J_c^k} \bar{c}^j(x) \quad \text{with } J_c^k \subset \{1, 2, \dots, k\}. \quad (4.2.7b)$$

It follows from convexity that

$$\check{f}^k(x) \leq f(x) \quad \text{and} \quad \check{c}^k(x) \leq c(x) \quad \text{for all } k \text{ and for all } x \in X.$$

Given a lower bound f_{low}^k for f_{min} , we define the so called improvement function at iteration k by

$$h^k(x) := \max\{f(x) - f_{\text{low}}^k, c(x)\}. \quad (4.2.8)$$

As already mentioned, the above definition of improvement function is not possible for proximal bundle methods, since the lower bounds f_{low}^k are not available. Some alternatives for f_{low}^k must be taken instead.

We have the following important result.

Lemma 4.2.3. *Consider the improvement function h^k given in (4.2.8). Then, $h^k(x) \geq 0$ for all $x \in X$ and for each iteration k . If there exists a sequence $\{x_{\text{rec}}^k\} \subset X$ such that $\lim_k h^k(x_{\text{rec}}^k) = 0$, then any cluster point of the sequence $\{x_{\text{rec}}^k\}$ is a solution to problem (4.2.2). In particular, if $h^k(x_{\text{rec}}^k) = 0$ for some k then x_{rec}^k is a solution to problem (4.2.2).*

Proof. Let $x \in X$ and k be arbitrary. If $c(x) \leq 0$, then $f(x) \geq f_{\text{low}}^k$ by definition of f_{low}^k ; on the other hand, if $f(x) < f_{\text{low}}^k$ then x cannot be feasible, i.e., $c(x) > 0$. It thus follows trivially that $h^k(x) \geq 0$ for all $x \in X$, as x and k were taken arbitrarily.

Assume that a sequence $\{x_{\text{rec}}^k\} \subset X$ such that $\lim_k h^k(x_{\text{rec}}^k) = 0$ is given. It follows from (4.2.8) that $h^k(x_{\text{rec}}^k) \geq f(x_{\text{rec}}^k) - f_{\text{low}}^k \geq f(x_{\text{rec}}^k) - f_{\text{min}}$. We can thus conclude (i) $0 \geq \lim_k (f(x_{\text{rec}}^k) - f_{\text{low}}^k) \geq \lim_k (f(x_{\text{rec}}^k) - f_{\text{min}})$ and similarly (ii) $0 \geq \lim_k c(x_{\text{rec}}^k)$.

Let \bar{x} be a cluster point of the sequence $\{x_{\text{rec}}^k\} \subset X$. Then $\bar{x} \in X$ and by (ii) $c(\bar{x}) \leq 0$. Therefore, since \bar{x} is feasible, by (i) we have that $0 \geq \lim_k (f(x_{\text{rec}}^k) - f_{\text{min}}) = f(\bar{x}) - f_{\text{min}} \geq 0$. Hence, \bar{x} is a solution to problem (4.2.2).

In particular, if $h^k(x_{\text{rec}}^k) = 0$ for some k then $0 \geq f(x_{\text{rec}}^k) - f_{\text{min}}$ and $0 \geq c(x_{\text{rec}}^k)$, showing that x_{rec}^k is a solution to problem (4.2.2). \square

That having been said, we will provide an algorithm that generates a sequence of iterates $\{x^k\} \subset X$, a subsequence of recorded iterates $\{x_{\text{rec}}^k\} \subset \{x^k\}$, a nondecreasing sequence of lower bounds $\{f_{\text{low}}^k\}$ for f_{min} , and a sequence of levels $f_{\text{lev}}^k \geq f_{\text{low}}^k$ such that

$$f_{\text{low}}^k \uparrow f_{\text{min}}, \quad f_{\text{lev}}^k \rightarrow f_{\text{min}} \quad \text{and} \quad h^k(x_{\text{rec}}^k) \rightarrow 0.$$

In order to do so, at iteration k we obtain a new iterate x^{k+1} by projecting a given stability center $\hat{x}^k \in X$ (not necessary feasible for c) onto the level set

$$\mathbb{X}^k := \{x \in X : \check{f}^k(x) \leq f_{\text{lev}}^k, \check{c}^k(x) \leq 0\} \quad (4.2.9)$$

whenever it is nonempty. Obtaining x^{k+1} amounts to solving the following quadratic problem (QP)

$$x^{k+1} := \arg \min_{x \in \mathbb{X}^k} \frac{1}{2} |x - \hat{x}^k|^2, \quad \text{or just } x^{k+1} := P_{\mathbb{X}^k}(\hat{x}^k) \text{ for short.} \quad (4.2.10)$$

Whenever \mathbb{X}^k is empty, the current level f_{lev}^k and lower bound f_{low}^k require updating.

The following proposition provides us with properties of the minimizer x^{k+1} . We will use the notation N_X for the normal cone of convex analysis of a set X .

Proposition 4.2.4. *The point x^{k+1} solves (4.2.10) if and only if $x^{k+1} \in X$, $\check{f}^k(x^{k+1}) \leq f_{\text{lev}}^k$, $\check{c}^k(x^{k+1}) \leq 0$, and there exist vectors $\hat{g}_f^k \in \partial \check{f}^k(x^{k+1})$, $\hat{g}_c^k \in \partial \check{c}^k(x^{k+1})$, $s^k \in N_X(x^{k+1})$ and stepsizes $\mu_f^k, \mu_c^k \geq 0$, such that*

$$x^{k+1} = \hat{x}^k - (\mu_f^k \hat{g}_f^k + \mu_c^k \hat{g}_c^k + s^k), \quad \mu_f^k (\check{f}^k(x^{k+1}) - f_{\text{lev}}^k) = 0 \quad \text{and} \quad \mu_c^k \check{c}^k(x^{k+1}) = 0. \quad (4.2.11)$$

In addition, the aggregate linearizations

$$\bar{f}^{a_f(k)}(\cdot) := \check{f}^k(x^{k+1}) + \langle \hat{g}_f^k, \cdot - x^{k+1} \rangle \quad \text{satisfies} \quad \bar{f}^{a_f(k)}(x) \leq \check{f}^k(x) \leq f(x) \quad \text{for all } x \in X \quad (4.2.12a)$$

$$\bar{c}^{a_c(k)}(\cdot) := \check{c}^k(x^{k+1}) + \langle \hat{g}_c^k, \cdot - x^{k+1} \rangle \quad \text{satisfies} \quad \bar{c}^{a_c(k)}(x) \leq \check{c}^k(x) \leq c(x) \quad \text{for all } x \in X. \quad (4.2.12b)$$

Moreover,

$$P_{\mathbb{X}^k}(\hat{x}^k) = P_{\mathbb{X}^{a(k)}}(\hat{x}^k) \quad (4.2.13)$$

holds, where $\mathbb{X}^{a(k)}$ is the aggregate level set defined as $\mathbb{X}^{a(k)} := \{x \in X : \bar{f}^{a_f(k)}(x) \leq f_{\text{lev}}^k, \bar{c}^{a_c(k)}(x) \leq 0\}$.

Proof. Let i_X be the indicator function of the polyhedral set X , i.e., $i_X(x) = 0$ if $x \in X$ and $i_X(x) = \infty$ otherwise. Remembering that the set $X \neq \emptyset$ is polyhedral and by [197, p.215] $\partial i_X(x) = N_X(x)$ for $x \in X$, the first claim results from the KKT conditions for problem (4.2.10) rewritten as

$$\begin{cases} \min_{x \in \mathbb{R}^n} & \frac{1}{2} |x - \hat{x}^k|^2 + i_X(x) \\ \text{s.t.} & \check{f}^k(x) \leq f_{\text{lev}}^k \\ & \check{c}^k(x) \leq 0. \end{cases}$$

The subgradient inequality gives (4.2.12).

We now proceed to show (4.2.13). Notice that $P_{\mathbb{X}^{a(k)}}(\hat{x}^k)$ is the solution to the following QP:

$$\begin{cases} \min_{x \in \mathbb{R}^n} & \frac{1}{2} |x - \hat{x}^k|^2 + i_X(x) \\ \text{s.t.} & \bar{f}^{a_f(k)}(x) \leq f_{\text{lev}}^k \\ & \bar{c}^{a_c(k)}(x) \leq 0. \end{cases} \quad (4.2.14)$$

Moreover, \bar{x} solves the above problem if, and only if, there exist $\rho, \lambda \geq 0$ and $\bar{s} \in N_X(\bar{x})$ such that

$$\bar{x} = \hat{x}^k - (\rho \hat{g}_f^k + \lambda \hat{g}_c^k + \bar{s}), \quad \rho (\bar{f}^{a_f(k)}(\bar{x}) - f_{\text{lev}}^k) = 0 \quad \text{and} \quad \lambda \bar{c}^{a_c(k)}(\bar{x}) = 0.$$

Notice that by taking $\rho = \mu_{\mathbf{f}}^k$, $\lambda = \mu_{\mathbf{c}}^k$ and $\bar{s} = s^k$ above, the resulting $\bar{x} = x^{k+1}$ satisfies the optimality conditions for problem (4.2.14). Since (4.2.14) has a unique solution, we conclude that $x^{k+1} = P_{\mathbb{X}^{a(k)}}(\hat{x}^k)$, as stated. \square

The interest of the aggregate level set is that $\mathbb{X}^{a(k)}$ condenses all the past information relevant for defining x^{k+1} . This property is crucial for the bundle compression mechanism. Indeed, this property makes it possible to change the feasible set \mathbb{X}^k in (4.2.10), possibly with many constraints, by any smaller set containing $\mathbb{X}^{a(k)}$. In this manner, the size of the bundles $J_{\mathbf{f}}^k$ and $J_{\mathbf{c}}^k$ in (4.2.7) can be kept controlled, thus making (4.2.10) easier to solve; see comment (g) below.

We are now in position to give the algorithm with full details. In the algorithm we will keep track of special iterations called “critical iterations” that allow us to either progress significantly in the quest of solving (4.2.2) or improve our currently proven lower bound for f_{\min} in (4.2.2). Critical iterations are counted with l . All iterations are counted with k . Iterations between the change of the critical iteration counter from l to $l+1$ are said to belong to the l -th cycle. We will moreover denote with $k(l)$ the critical iteration at the start of the l -th cycle and hence $K^l = \{k(l), k(l)+1, \dots, k(l+1)-1\}$ is the index set gathering the iterations in the l -th cycle. The recorded sequence $\{x_{\text{rec}}^k\}$ is constructed by defining x_{rec}^0 equal to the first iterate x^0 and

$$x_{\text{rec}}^k := \begin{cases} x_{\text{rec}}^{k-1} & \text{if } \begin{cases} \min_{j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k} h^k(x^j) > h^{k-1}(x_{\text{rec}}^{k-1}) \text{ and } k \geq 1 \\ \text{or } \{J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k\} = \emptyset, \end{cases} \\ \arg \min_{j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k} h^k(x^j) & \text{otherwise.} \end{cases} \quad (4.2.15)$$

Algorithm 4.2 (Level Bundle Method). *We assume given an oracle computing f/c values as in (4.2.6) for any $x \in X$.*

Step 0 (Input and Initialization) *Choose a parameter $\gamma \in (0, 1)$ and a stopping tolerance $\delta_{\text{Tot}} > 0$. Given $x^0 \in X$ and a lower bound $f_{\text{low}}^0 \leq f_{\min}$, set $\hat{x}^0 \leftarrow x^0$. Call the oracle to obtain $(f(x^0), g_{\mathbf{f}}^0)$ and $(c(x^0), g_{\mathbf{c}}^0)$ and set $l \leftarrow 0$, $k \leftarrow 0$, $k(l) \leftarrow 0$, $J_{\mathbf{f}}^0 \leftarrow \{0\}$ and $J_{\mathbf{c}}^0 \leftarrow \{0\}$.*

Step 1 (Best Current Minimizer) *Update the recorded sequence x_{rec}^k according to (4.2.15).*

Step 2 (Stopping Test) *Perform the stopping test $h^k(x_{\text{rec}}^k) \leq \delta_{\text{Tot}}$. If satisfied return x_{rec}^k and $f(x_{\text{rec}}^k)$.*

Step 3 (Descent Test) *In this optional Step we test if $h^k(x_{\text{rec}}^k) \leq (1 - \gamma)h^{k(l)}(x_{\text{rec}}^{k(l)})$. If so we declare a critical iteration and we set $l \leftarrow l + 1$, $k(l) \leftarrow k$ and choose $\hat{x}^k \in \{x^j : j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k\}$.*

Step 4 (Level Updating) *Set $f_{\text{lev}}^k \leftarrow f_{\text{low}}^k + \gamma h^k(x_{\text{rec}}^k)$ and update the feasible set \mathbb{X}^k given in (4.2.9).*

Step 5 (Quadratic Program) *Try to solve the quadratic program (4.2.10). If no feasible solution is found then declare a critical iteration, set $l \leftarrow l + 1$, $k(l) \leftarrow k$, $f_{\text{low}}^k \leftarrow f_{\text{lev}}^k$ and choose $\hat{x}^k \in \{x^j : j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k\}$. Return to Step 1. If problem (4.2.10) is solved then move to Step 6.*

Step 6 (Oracle) Call the oracle to obtain $(f(x^{k+1}), g_{\mathbf{f}}^{k+1})$ and $(c(x^{k+1}), g_c^{k+1})$.

Set $f_{\text{low}}^{k+1} \leftarrow f_{\text{low}}^k$ and $\hat{x}^{k+1} \leftarrow \hat{x}^k$.

Step 7 (Bundle Management) Manage the bundle freely as long as $J_{\mathbf{f}}^{k+1} \supset \{k+1, a_{\mathbf{f}}(k)\}$ and $J_c^{k+1} \supset \{k+1, a_c(k)\}$ remain true at each bundle compression.

Step 8 (Loop) Set $k = k+1$ and go to Step 1.

We can provide the following remarks concerning Algorithm 4.2:

- (a) The sequence $\{h^k(x_{\text{rec}}^k)\}$ is monotonically decreasing by (4.2.8) and (4.2.15).
- (b) Let K^l be the index set belonging to the l -th cycle, it then follows that both the stability center \hat{x}^k and lower bound f_{low}^k are fixed for all $k \in K^l$. As a consequence,

$$\text{for each fixed } l \geq 0, \text{ the sequence } \{f_{\text{lev}}^j\}_{j \in K^l} \text{ is nonincreasing.} \quad (4.2.16)$$

This property is essential for Lemma 4.2.7.

It may happen by Step 3 and 5 that K^l is an empty set. In the convergence analysis given below we simply assume that K^l is nonempty for each l . This strategy eases the calculations without loosing generality.

- (c) If the level set \mathbb{X}^k is empty, this means that $f_{\text{lev}}^k < \check{f}^k(x)$ for all $x \in \{z \in X : \check{c}^k(z) \leq 0\}$. Since $\check{c}^k(\cdot) \leq c(\cdot)$, it follows that $\{z \in X : c(z) \leq 0\} \subseteq \{z \in X : \check{c}^k(z) \leq 0\}$ and thus, $f_{\text{lev}}^k < \check{f}^k(x)$ for all $x \in \{z \in X : c(z) \leq 0\}$. As $\check{f}^k(\cdot) \leq f(\cdot)$ we conclude that f_{lev}^k is a lower bound for the optimal value f_{min} . Therefore, the updating rule $f_{\text{low}}^k \leftarrow f_{\text{lev}}^k$ at Step 4 gives a valid lower bound for f_{min} .
- (d) Whenever the level set is empty at Step 5, the lower bound f_{low}^k is increased by at least an amount of $\gamma h^k(x_{\text{rec}}^k) > 0$ and the cycle-counter l is incremented by one. Moreover $f_{\text{low}}^k \leq f_{\text{min}}$ for all $k \geq 0$. Assume that $\delta_{\text{To1}} = 0$. If there is an infinite loop between Steps 5 and 1, one concludes that $f_{\text{low}}^{k(l+1)} > f_{\text{low}}^{k(l)}$ because the stopping test in Step 2 fails. Since $f_{\text{min}} < \infty$ is an upper bound on this sequence, we must therefore have that $h^k(x_{\text{rec}}^k) \rightarrow 0$. Hence, if $\delta_{\text{To1}} > 0$ there cannot be an infinite loop between Steps 5 and 1.
- (e) In order to avoid excessively short cycles, we can solve the following linear problem to update f_{low}^k at Step 5:

$$f_{\text{low}}^k = \min_{x \in X} \check{f}^k(x) \quad \text{s.t.} \quad \check{c}^k(x) \leq 0. \quad (4.2.17)$$

As $f_{\text{lev}}^k > f_{\text{low}}^k$, then \mathbb{X}^k will be nonempty.

- (f) If problem (4.2.17) is solved at each iteration at Step 3, then the level set \mathbb{X}^k is nonempty. However, this procedure might not ensure (4.2.16) and thus Lemma 4.2.7 below holds only if there is no bundle compression along each cycle K^l , i.e., $J_{\mathbf{f}}^{k+1} = J_{\mathbf{f}}^k \cup \{k+1\}$ and $J_c^{k+1} = J_c^k \cup \{k+1\}$ for all $k \in K^l$. For more details see [136, p. 121] or [44, Lemma 3.5].
- (g) In fact, the aggregate linearizations (4.2.12) do not need to be included in the bundle at each iteration but only when the bundle is compressed. This provides a versatile framework for dealing with bundle compression and avoid encumbering memory. Considering that the bundle is full (we have saved more elements than some a priori

bound), a quite practical rule would consist of keeping only the active linearization index sets:

$$\bar{J}_f^k := \{j \in J_f^k : \bar{f}^j(x^{k+1}) = f_{\text{lev}}^k\} \quad \text{and} \quad \bar{J}_c^k := \{j \in J_c^k : \bar{c}^j(x^{k+1}) = 0\}.$$

If the bundle is still considered full, we can replace any two indices $i, j \in \bar{J}_f^k$ by $a_f(k)$ (respectively $i, j \in \bar{J}_c^k$ by $a_c(k)$) until the Bundle is sufficiently cleaned up. Then we update as follows:

$$J_f^{k+1} = \bar{J}_f^k \cup \{k+1\} \quad \text{and} \quad J_c^{k+1} = \bar{J}_c^k \cup \{k+1\}.$$

(h) After having solved problem (4.2.10), each constraint $\bar{f}^j(x^{k+1}) \leq f_{\text{lev}}^k$ with $j \in J_f^k$ (respec. $\bar{c}^i(x^{k+1}) \leq 0$ with $i \in J_c^k$) has a Lagrange multiplier $\alpha_f^j \geq 0$ (respec. $\alpha_c^i \geq 0$). Furthermore, it follows from the cutting-plane definition that $g_f^j \in \partial f(x^{k+1})$ for all $j \in J_f^k$ such that $\alpha_f^j > 0$ (respec. $g_c^i \in \partial c(x^{k+1})$ for all $i \in J_c^k$ such that $\alpha_c^i > 0$). In this manner, by taking $\mu_f^k = \sum_{j \in J_f^k} \alpha_f^j$ and $\hat{g}_f^k = \frac{1}{\mu_f^k} \sum_{j \in J_f^k} \alpha_f^j g_f^j$ (respec. $\mu_c^k = \sum_{j \in J_c^k} \alpha_c^j$ and $\hat{g}_c^k = \frac{1}{\mu_c^k} \sum_{j \in J_c^k} \alpha_c^j g_c^j$) the aggregate linearizations (4.2.12) can be easily obtained. If $\mu_f^k = 0$, then all constraints $\bar{f}^j(x^{k+1}) \leq f_{\text{lev}}^k$ are inactive (the inequality holds strictly) and all the bundle J_f^k information is useless. Then the aggregate linearization $\bar{f}^{a(k)}$ is also inactive. The equivalent conclusion holds if $\mu_c^k = 0$. Lemma 4.2.6 below ensures that $\mu_c^k = \mu_f^k = 0$ cannot happen, otherwise we would have $x^{k+1} = \hat{x}^k$.

Steps 6 and 7 in Algorithm 4.2 are optional in a way. We can consider two important alternatives which will be particularly useful when either the f or the c oracle is costly.

A variant of Algorithm 4.2 for costly c -oracles

The following strategy is useful if the constraint oracle information $(c(x^{k+1}), g_c^{k+1})$ is more difficult to obtain than the information $(f(x^{k+1}), g_f^{k+1})$. This is the case of Example 4.2.2.

Goal Avoid computing the c -oracle information if x^{k+1} does not give a significant decrease.

Step 6(f) (Oracle) Set

$$f_{\text{low}}^{k+1} \leftarrow f_{\text{low}}^k, \quad \hat{x}^{k+1} \leftarrow \hat{x}^k$$

and call the oracle to obtain $(f(x^{k+1}), g_f^{k+1})$. If $f(x^{k+1}) \geq f_{\text{lev}}^k + (1 - \gamma)h^k(x_{\text{rec}}^k)$ go to Step 7(f). Otherwise, before going to Step 7(f) compute $(c(x^{k+1}), g_c^{k+1})$.

Step 7(f) (Bundle Management) Manage the bundle freely as long as:

- (If both $f(x^{k+1})$ and $c(x^{k+1})$ were computed at Step 6(f))
 $J_f^{k+1} \supset \{k+1, a_f(k)\}$ and $J_c^{k+1} \supset \{k+1, a_c(k)\}$ hold true at each bundle compression.
- (If $c(x^{k+1})$ was not computed at Step 6(f))
 $J_f^{k+1} \supset \{k+1, a_f(k)\}$ and $J_c^{k+1} \supset \{a_c(k)\}$ hold true at each bundle compression.

We emphasize that the test $f(x^{k+1}) \geq f_{\text{lev}}^k + (1 - \gamma)h^k(x_{\text{rec}}^k)$ implies that the c -oracle information is computed only if $f(x^{k+1}) < \max\{f(x_{\text{rec}}^k), f_{\text{low}}^k + c(x_{\text{rec}}^k)\}$, i.e., when $f(x^{k+1})$ provides decrease with respect to at least one of the two thresholds $f(x_{\text{rec}}^k)$ or $f_{\text{low}}^k + c(x_{\text{rec}}^k)$.

A variant of Algorithm 4.2 for costly f -oracles

If $(f(x^{k+1}), g_{\mathbf{f}}^{k+1})$ is more difficult to compute than $(c(x^{k+1}), g_{\mathbf{c}}^{k+1})$ we may consider the following strategy in Algorithm 4.2:

Goal Avoid computing the f -oracle information if x^{k+1} is far from feasibility.

Step 6(c) (Oracle) Set

$$f_{\mathbf{low}}^{k+1} \leftarrow f_{\mathbf{low}}^k, \hat{x}^{k+1} \leftarrow \hat{x}^k$$

and call the oracle to obtain $(c(x^{k+1}), g_{\mathbf{c}}^{k+1})$. If $c(x^{k+1}) \geq (1 - \gamma)h^k(x_{\mathbf{rec}}^k)$ go to Step 7(c). Otherwise, before going to Step 7(c) compute $(f(x^{k+1}), g_{\mathbf{f}}^{k+1})$.

Step 7(c) (Bundle Management) Manage the bundle freely as long as:

- (If both $f(x^{k+1})$ and $c(x^{k+1})$ were computed at Step 6(c))
 $J_{\mathbf{f}}^{k+1} \supset \{k+1, a_{\mathbf{f}}(k)\}$ and $J_{\mathbf{c}}^{k+1} \supset \{k+1, a_{\mathbf{c}}(k)\}$ remain true at each bundle compression.
- (If $f(x^{k+1})$ was not computed at Step 6(c))
 $J_{\mathbf{c}}^{k+1} \supset \{k+1, a_{\mathbf{c}}(k)\}$ and $J_{\mathbf{f}}^{k+1} \supset \{a_{\mathbf{f}}(k)\}$ remain true at each bundle compression.

According to the above two strategies, we consider the following three versions for Algorithm 4.2:

Algorithm 4.2(fc) Algorithm 4.2 as it is stated;

Algorithm 4.2(f) Algorithm 4.2 with Steps 6 and 7 replaced by Steps 6(f) and 7(f);

Algorithm 4.2(c) Algorithm 4.2 with Steps 6 and 7 replaced by Steps 6(c) and 7(c).

Remark 4.2.5. Notice that the index set $J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$ can be empty for versions (f) and (c) of Algorithm 4.2. If it is the case, the choice of the next stability center \hat{x}^k at Steps 3 and 5 is not well defined. We may proceed as follows according to Steps 6(f) and 6(c):

- At Steps 3 and 5 of Algorithm 4.2(f), choose $\hat{x}^k \in \{x^j : j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k\}$ if $J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k \neq \emptyset$.
 Otherwise, $\hat{x}^k \in \{x^j : j \in J_{\mathbf{f}}^k \text{ and } f(x^j) \geq f_{\mathbf{lev}}^{j-1} + (1 - \gamma)h^{j-1}(x_{\mathbf{rec}}^{j-1})\}$.
- At Steps 3 and 5 of Algorithm 4.2(c), choose $\hat{x}^k \in \{x^j : j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k\}$ if $J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k \neq \emptyset$.
 Otherwise, $\hat{x}^k \in \{x^j : j \in J_{\mathbf{c}}^k \text{ and } c(x^j) \geq (1 - \gamma)h^{j-1}(x_{\mathbf{rec}}^{j-1})\}$.

Here comes an interesting feature of Algorithm 4.2: differently from the constrained level bundle methods [136], [127], [70], [71] and version (fc), the versions (f) and (c) of Algorithm 4.2 do not need to include the two linearizations \bar{f}^{k+1} and \bar{c}^{k+1} into the bundle at each iteration k . As shown in Theorem 4.2.9 below, even considering less oracle information, versions (f) and (c) provide the same complexity result than version (fc).

Convergence analysis of the above three versions ((fc), (f) and (c)) of Algorithm 4.2 is given in the following subsection.

4.2.2.1 Convergence analysis

Throughout this section we suppose the feasible set X in (4.2.2) is compact with diameter D , and denote by $\Lambda^f > 0$ (respectively Λ^c) a Lipschitz constant for the objective function

(respectively constraint) over the set X . Therefore, $\Lambda := \max\{\Lambda^f, \Lambda^c\}$ is a Lipschitz constant for h^k , defined in (4.2.8). In what follows, both constants D and Λ can be unknown, since they are not used by the algorithm but only to give upper bounds for the length of cycles K^l .

In the following lemma, we extend the result given in [127, Lemma 3.1] to our setting.

Lemma 4.2.6. *Let $l \geq 0$ be arbitrary, $\Lambda = \max\{\Lambda^f, \Lambda^c\}$ and $\gamma \in (0, 1)$ be given. At iteration k of any version (fc), (f) or (c) of Algorithm 4.2, the following estimates hold:*

$$\begin{aligned} |x^{k+1} - x^k| &\geq \frac{(1-\gamma)}{\Lambda} h^k(x_{\text{rec}}^k) \quad \text{if } k > k(l) \\ |x^{k+1} - \hat{x}^k| &\geq \frac{(1-\gamma)}{\Lambda} h^k(x_{\text{rec}}^k) \quad \text{if } k = k(l). \end{aligned}$$

Proof. Let us split this proof in three cases: we consider Algorithm 4.2(fc) first, then Algorithm 4.2(f) and finally Algorithm 4.2(c). We now study the first case.

Let k be arbitrary and $j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$ be given. By (4.2.9) combined with (4.2.10), we get

$$\begin{aligned} f(x^j) + \langle g_{\mathbf{f}}^j, x^{k+1} - x^j \rangle &\leq f_{\mathbf{lev}}^k \\ c(x^j) + \langle g_{\mathbf{c}}^j, x^{k+1} - x^j \rangle &\leq 0. \end{aligned}$$

By applying the Cauchy-Schwarz inequality we thus derive

$$\begin{aligned} f(x^j) - f_{\mathbf{lev}}^k &\leq |g_{\mathbf{f}}^j| |x^{k+1} - x^j| \leq \Lambda^f |x^{k+1} - x^j| \\ c(x^j) &\leq |g_{\mathbf{c}}^j| |x^{k+1} - x^j| \leq \Lambda^c |x^{k+1} - x^j|. \end{aligned}$$

Since $\Lambda = \max\{\Lambda^f, \Lambda^c\}$, $f_{\mathbf{lev}}^k = f_{\mathbf{low}}^k + \gamma h^k(x_{\text{rec}}^k)$ and $h^k(x_{\text{rec}}^k) \geq 0$, we conclude that

$$\begin{aligned} \Lambda |x^{k+1} - x^j| &\geq \max\{f(x^j) - f_{\mathbf{low}}^k - \gamma h^k(x_{\text{rec}}^k), c(x^j)\} \\ &\geq \max\{f(x^j) - f_{\mathbf{low}}^k - \gamma h^k(x_{\text{rec}}^k), c(x^j) - \gamma h^k(x_{\text{rec}}^k)\} \\ &= -\gamma h^k(x_{\text{rec}}^k) + \max\{f(x^j) - f_{\mathbf{low}}^k, c(x^j)\} \\ &= -\gamma h^k(x_{\text{rec}}^k) + h^k(x^j) \\ &\geq (1-\gamma) h^k(x_{\text{rec}}^k). \end{aligned} \tag{4.2.18}$$

Assuming that $k > k(l)$, then the Oracle and Bundle Management steps assure that $k \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$. Hence the estimate (4.2.18) with $j = k$ provides the result. When $k = k(l)$, the stability center \hat{x}^k is chosen among the bundle members. Hence, there exists a $j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$ such that $x^j = \hat{x}^k$ and (4.2.18) allows us to conclude this proof for the first case.

For versions 4.2(f) and 4.2(c) of Algorithm, if $k \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$ or $j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$ with $x^j = \hat{x}^k$ nothing more needs to be shown. Consider Algorithm 4.2(f) and $j \in J_{\mathbf{f}}^k$ but $j \notin J_{\mathbf{c}}^k$ together with $f(x^j) - f_{\mathbf{lev}}^{j-1} \geq (1-\gamma) h^{j-1}(x_{\text{rec}}^{j-1})$ which holds for $j = k$ and j s.t. $x^j = \hat{x}^k$. As $f_{\mathbf{lev}}^{j-1} \geq f_{\mathbf{lev}}^k$ by (4.2.16) and $h^{j-1}(x_{\text{rec}}^{j-1}) \geq h^k(x_{\text{rec}}^k)$ by (4.2.15) and comment (a), it follows that

$$f(x^j) - f_{\mathbf{lev}}^k \geq (1-\gamma) h^k(x_{\text{rec}}^k).$$

Since x^{k+1} is feasible for (4.2.10), then $f(x^j) + \langle g_f^j, x^{k+1} - x^j \rangle \leq f_{\text{lev}}^k$. All together we conclude that

$$\Lambda |x^{k+1} - x^j| \geq \langle g_f^j, x^j - x^{k+1} \rangle \geq f(x^j) - f_{\text{lev}}^k \geq (1 - \gamma)h^k(x_{\text{rec}}^k),$$

which was to be shown.

Consider Algorithm 4.2(c) and $j \in J_c^k$ but $j \notin J_f^k$ such that $c(x^j) \geq (1 - \gamma)h^{j-1}(x_{\text{rec}}^{j-1})$, which holds for $j = k$ and j s.t. $x^j = \hat{x}^k$. In this case we obtain $c(x^j) \geq (1 - \gamma)h^{j-1}(x_{\text{rec}}^{j-1}) \geq (1 - \gamma)h^k(x_{\text{rec}}^k)$, where the last inequality follows from (4.2.15). Since x^{k+1} is feasible for (4.2.10), then $0 \geq c(x^j) + \langle g_c^j, x^{k+1} - x^j \rangle$. It follows from Cauchy-Schwarz inequality that

$$\Lambda |x^{k+1} - x^j| \geq \langle g_c^j, x^j - x^{k+1} \rangle \geq c(x^j) \geq (1 - \gamma)h^k(x_{\text{rec}}^k),$$

which concludes the proof. \square

Lemma 4.2.6 shows that consecutive iterates are different. The following result implies that each cycle K^l is finite.

Lemma 4.2.7. *Let $l \geq 0$ be arbitrary, D be the diameter of the feasible set X and $\Lambda = \max\{\Lambda^f, \Lambda^c\}$. Assume that at each bundle compression in the versions (fc), (f) and (c) of Algorithm 4.2 we include the aggregate indices in the bundle during the l -th cycle. Then, for all three versions (fc), (f) and (c) of Algorithm 4.2, any iteration k in the l -th cycle (i.e., $k \in K^l$) with $h^k(x_{\text{rec}}^k) > \delta_{\text{Tot}}$ may differ no more from $k(l)$ than the following bound:*

$$k - k(l) + 1 \leq \left(\frac{\Lambda D}{(1 - \gamma)h^k(x_{\text{rec}}^k)} \right)^2$$

Proof. Let $k \in K^l$ be such that $k > k(l)$. It follows from (4.2.10) that $x^k = P_{\mathbb{X}^{k-1}}(\hat{x}^{k-1})$; thus

$$\langle \hat{x}^{k-1} - x^k, x - x^k \rangle \leq 0 \text{ for all } x \in \mathbb{X}^{k-1}. \quad (4.2.19)$$

If no bundle compression took place between step $k - 1$ and step k , it holds that $\check{f}^k(x) \geq \check{f}^{k-1}(x)$, $\check{c}^k(x) \geq \check{c}^{k-1}(x)$ for all $x \in \mathbb{R}^n$. Since $f_{\text{lev}}^k \leq f_{\text{lev}}^{k-1}$ by (4.2.16), it follows that $\mathbb{X}^k \subseteq \mathbb{X}^{k-1}$. Since k belongs to the l -th cycle, \mathbb{X}^k is nonempty and $x^{k+1} \in \mathbb{X}^k$. This fact combined with $\hat{x}^{k-1} = \hat{x}^k$ and (4.2.19) allows us to deduce that $\langle \hat{x}^k - x^k, x^{k+1} - x^k \rangle \leq 0$. When bundle compression took place between steps $k - 1$ and k , the aggregate indices belong to the bundle for each of the three considered versions, i.e., $a_f(k - 1) \in J_f^k$ and $a_c(k - 1) \in J_c^k$ for Algorithms 4.2(fc), 4.2(f) and 4.2(c). This implies $\check{f}^k(x) \geq \bar{f}^{a_f(k-1)}(x)$, $\check{c}^k(x) \geq \bar{c}^{a_c(k-1)}(x)$ for all $x \in \mathbb{R}^n$. It follows from (4.2.13) that $x^k = P_{\mathbb{X}^{a(k-1)}}(\hat{x}^{k-1})$ and thus,

$$\langle \hat{x}^{k-1} - x^k, x - x^k \rangle \leq 0 \text{ for all } x \in \mathbb{X}^{a(k-1)}. \quad (4.2.20)$$

Since $f_{\text{lev}}^k \leq f_{\text{lev}}^{k-1}$ by (4.2.16), it follows that $\mathbb{X}^k \subset \mathbb{X}^{a(k-1)}$. Similarly as before we can use (4.2.20) to obtain $\langle \hat{x}^k - x^k, x^{k+1} - x^k \rangle \leq 0$ for each of the versions (fc), (f) and (c). Therefore, by developing squares in the identity $|x^{k+1} - \hat{x}^k|^2 = |x^{k+1} - x^k + (x^k - \hat{x}^k)|^2$ we have that

$$|x^{k+1} - \hat{x}^k|^2 \geq |x^k - \hat{x}^k|^2 + |x^{k+1} - x^k|^2.$$

As $h^k(x_{\text{rec}}) > \delta_{\text{Tot}}$ for all $k \in K^l$, Algorithm 4.2 does not stop and $\hat{x}^k = \hat{x}^{k(l)}$. A recursive application of the above inequality implies

$$\begin{aligned} |x^{k+1} - \hat{x}^k|^2 &\geq |x^k - \hat{x}^k|^2 + |x^{k+1} - x^k|^2 \\ &= |x^k - \hat{x}^{k-1}|^2 + |x^{k+1} - x^k|^2 \\ &\geq |x^{k-1} - \hat{x}^{k-1}|^2 + |x^k - x^{k-1}|^2 + |x^{k+1} - x^k|^2 \\ &\vdots \\ &\geq |x^{k(l)+1} - \hat{x}^{k(l)}|^2 + \sum_{j=k(l)+1}^k |x^{j+1} - x^j|^2. \end{aligned}$$

Since X is compact with diameter D and both x^{k+1} and \hat{x}^k are in X , then $D \geq |x^{k+1} - \hat{x}^k|$. Together with Lemma 4.2.6, this yields the bounds

$$D^2 \geq \left(\frac{(1-\gamma)}{\Lambda} h^k(x_{\text{rec}}^k) \right)^2 + \sum_{j=k(l)+1}^k \left(\frac{(1-\gamma)}{\Lambda} h^k(x_{\text{rec}}^k) \right)^2 = \left(\frac{(1-\gamma)}{\Lambda} h^k(x_{\text{rec}}^k) \right)^2 (k - k(l) + 1),$$

and result follows as k was arbitrary. \square

If the stopping test tolerance is set to zero, we now show that any cluster point of the recorded sequence of points generated by any version of Algorithm 4.2 solves problem (4.2.2).

Theorem 4.2.8. *Consider any version (fc), (f) or (c) of Algorithm 4.2. Suppose that $\delta_{\text{Tot}} = 0$ and that the algorithm does not terminate. Then $\lim_k h^k(x_{\text{rec}}^k) = 0$ and any cluster point of the sequence $\{x_{\text{rec}}^k\}$ is a solution to problem (4.2.2).*

Proof. If any version of Algorithm 4.2 stops at iteration k with $h^k(x_{\text{rec}}^k) = 0$, by Lemma 4.2.3 x_{rec}^k is an optimal solution to problem (4.2.2).

If the algorithms do not stop, then by Lemma 4.2.7 (each cycle K^l is finite) we conclude that $l \rightarrow \infty$. Let \mathcal{W} (respec. \mathcal{L}) be an index set gathering counters l updated at Step 3 (respec. Step 5) of (any version of) Algorithm 4.2.

Let l_i be the i -th index in $\mathcal{W} = \{l_1, l_2, \dots\}$. Therefore, the inequality $h^{k(l_i)}(x_{\text{rec}}^{k(l_i)}) \leq (1-\lambda)h^{k(l_{i-1})}(x_{\text{rec}}^{k(l_{i-1})})$ holds even if there exists some $l \in \mathcal{L}$ such that $l_{i-1} < l < l_i$. A recursive application of the previous inequality gives us

$$0 \leq h^{k(l_i)}(x_{\text{rec}}^{k(l_i)}) \leq (1-\lambda)h^{k(l_{i-1})}(x_{\text{rec}}^{k(l_{i-1})}) \leq \dots \leq (1-\gamma)^i h^{k(0)}(x_{\text{rec}}^{k(0)}).$$

As both function f and c are finite valued, if \mathcal{W} has infinitely many indices then $\lim_i h^{k(l_i)}(x_{\text{rec}}^{k(l_i)}) = 0$. Moreover, $\lim_k h^k(x_{\text{rec}}^k) = 0$ by monotonicity; see (4.2.15).

Let us now suppose that \mathcal{W} is finite. As $l \rightarrow \infty$, then the index set \mathcal{L} has infinitely many indices. As $f_{\text{low}}^{k(l+1)}$ is increased at Step 5 of the algorithm by an amount of $\gamma h^k(x_{\text{rec}}^k) \geq 0$ for each $l \in \mathcal{L}$, we obtain that $\lim_k h^k(x_{\text{rec}}^k) = 0$ (otherwise we would have $f_{\text{low}}^k \uparrow \infty$, which contradicts $f_{\text{low}}^k \leq f_{\text{min}} < \infty$).

We have shown in both cases (\mathcal{W} finite or infinite) that $\lim_k h^k(x_{\text{rec}}^k) = 0$. Hence, the stated result follows from Lemma 4.2.3. \square

4.2.2.2 Complexity results

We now provide an upper bound for the maximum number of iterations performed by versions (fc), (f) and (c) of Algorithm 4.2 in order to reach a given tolerance $\delta_{\text{To1}} > 0$. To do so, we consider all the versions of Algorithm 4.2 without the optional Step 3.

Theorem 4.2.9. *Let D be the diameter of the feasible set X and $\Lambda = \max\{\Lambda^f, \Lambda^c\}$. Let $-\infty < f_{\text{low}}^1 \leq f_{\text{min}}$ be given in all versions (fc), (f) and (c) of Algorithm 4.2 and let the Step 3 be deactivated. Assume that at each bundle compression the aggregate indices are included in the bundle. Then, to reach an optimality measure $h^k(x_{\text{rec}}^k)$ smaller than $\delta_{\text{To1}} > 0$ Algorithms 4.2(fc), 4.2(f) and 4.2(c) perform at most*

$$\left(1 + \frac{f_{\text{min}} - f_{\text{low}}^1}{\gamma \delta_{\text{To1}}}\right) \left(\frac{\Lambda D}{(1 - \gamma) \delta_{\text{To1}}}\right)^2 \text{ iterations.}$$

Proof. Consider any arbitrary but fixed version of Algorithm 4.2. Notice that every time that $\mathbb{X}^k = \emptyset$, the lower bound f_{low}^k for the optimal value f_{min} is increased by an amount of $\gamma h^k(x_{\text{rec}}^k)$ ($> \gamma \delta_{\text{To1}}$). Since f_{low}^1 is finite, the maximum number of cycles l^{mx} times the stepsize $\gamma \delta_{\text{To1}}$ is less than $f_{\text{min}} - f_{\text{low}}^1$, i.e.,

$$l^{\text{mx}} \leq \frac{f_{\text{min}} - f_{\text{low}}^1}{\gamma \delta_{\text{To1}}}.$$

It follows from Lemma 4.2.7 that each cycle K^l has at most $\left(\frac{\Lambda D}{(1 - \gamma) \delta_{\text{To1}}}\right)^2$ iterations, where we have used that $h^k(x_{\text{rec}}^k) > \delta_{\text{To1}}$ since the algorithm did not stop at iteration k . Let $K(\delta_{\text{To1}}) := \{1, 2, \dots\}$ be the index set for which $h^k(x_{\text{rec}}^k) > \delta_{\text{To1}}$. We have thus shown that $K(\delta_{\text{To1}})$ is finite and if $k_{\delta_{\text{To1}}}$ is its maximal element, then:

$$k_{\delta_{\text{To1}}} \leq \left(1 + \frac{f_{\text{min}} - f_{\text{low}}^1}{\gamma \delta_{\text{To1}}}\right) \left(\frac{\Lambda D}{(1 - \gamma) \delta_{\text{To1}}}\right)^2,$$

as was to be shown. □

Notice that the better is the initial lower bound f_{low}^1 for the optimal value f_{min} , the lower is the upper bound for the maximum number of iterations performed by any considered version of Algorithm 4.2. If a lower bound f_{low}^1 is not available, we might obtain an initial lower bound by solving a linear program, as follows.

Corollary 4.2.10. *In the setting of Theorem 4.2.9, suppose that the initial lower bound f_{low}^1 is defined as*

$$f_{\text{low}}^1 = \min \check{f}^1(x) \text{ s.t. } \check{c}^1(x) \leq 0, x \in X.$$

Then, $f_{\text{min}} - f_{\text{low}}^1 \leq 2\Lambda^f D$ and to reach an optimality measure $h^k(x_{\text{rec}}^k)$ smaller than $\delta_{\text{To1}} > 0$ Algorithms 4.2(fc), 4.2(f) and 4.2(c) perform at most

$$\left(1 + \frac{2\Lambda^f D}{\gamma \delta_{\text{To1}}}\right) \left(\frac{\Lambda D}{(1 - \gamma) \delta_{\text{To1}}}\right)^2 \text{ iterations.}$$

Proof. We have that $f_{\min} = f(x_{\min}) \leq f(x^1) + \Lambda^f D$. Let \tilde{x} be a solution to the above linear program. Then $f_{\text{low}}^1 = \tilde{f}^1(\tilde{x}) = f(x^1) + \langle g_{\mathbf{f}}^1, \tilde{x} - x^1 \rangle \geq f(x^1) - \Lambda^f D$ by Cauchy-Schwarz inequality. The result follows by combining the preceding inequalities and invoking Theorem 4.2.9. \square

So far we have considered only exact oracle information. In the next section we deal with inexactness from the oracle.

4.2.3 Level Bundle Method for Inexact Oracles

In this section we assume that both functions f and c in (4.2.2) are inexactly evaluated. In what follows we will assume that the oracle provides us for each $x \in X$ with

$$\left\{ \begin{array}{l} f\text{-oracle information} \left[\begin{array}{l} f_x = f(x) - \eta_{\mathbf{f}}^x \text{ and} \\ \tilde{g}_{\mathbf{f}}^x \in \mathbb{R}^n \text{ such that } f(\cdot) \geq f_x + \langle \tilde{g}_{\mathbf{f}}^x, \cdot - x \rangle - \epsilon_{\mathbf{f}}^x \end{array} \right. \\ \\ c\text{-oracle information} \left[\begin{array}{l} c_x = c(x) - \eta_{\mathbf{c}}^x \text{ and} \\ \tilde{g}_{\mathbf{c}}^x \in \mathbb{R}^n \text{ such that } c(\cdot) \geq c_x + \langle \tilde{g}_{\mathbf{c}}^x, \cdot - x \rangle - \epsilon_{\mathbf{c}}^x \end{array} \right. \end{array} \right. \quad (4.2.21)$$

for some unknown $\eta_{\mathbf{f}}^x, \eta_{\mathbf{c}}^x, \epsilon_{\mathbf{f}}^x, \epsilon_{\mathbf{c}}^x \in \mathbb{R}$.

By substituting $f_x = f(x) - \eta_{\mathbf{f}}^x$ in the second inequality and evaluating at x we derive that $\eta_{\mathbf{f}}^x \geq -\epsilon_{\mathbf{f}}^x$. Similarly we observe that $\eta_{\mathbf{c}}^x \geq -\epsilon_{\mathbf{c}}^x$. Therefore, $\tilde{g}_{\mathbf{f}}^x \in \partial_{(\eta_{\mathbf{f}}^x + \epsilon_{\mathbf{f}}^x)} f(x)$ and $\tilde{g}_{\mathbf{c}}^x \in \partial_{(\eta_{\mathbf{c}}^x + \epsilon_{\mathbf{c}}^x)} c(x)$.

Throughout this section we will make the assumption that the error on each of these estimates is bounded, i.e., there exist constants $\eta_{\mathbf{f}}, \eta_{\mathbf{c}}, \epsilon_{\mathbf{f}}, \epsilon_{\mathbf{c}} \geq 0$ such that

$$|\eta_{\mathbf{f}}^x| \leq \eta_{\mathbf{f}}, \epsilon_{\mathbf{f}}^x \leq \epsilon_{\mathbf{f}}, |\eta_{\mathbf{c}}^x| \leq \eta_{\mathbf{c}}, \epsilon_{\mathbf{c}}^x \leq \epsilon_{\mathbf{c}} \text{ for all } x \in X. \quad (4.2.22)$$

With such information from the oracle, we consider the approximate linearizations

$$\bar{f}^k(x) := f_x^k + \langle \tilde{g}_{\mathbf{f}}^k, x - x^k \rangle \text{ and } \bar{c}^k(x) := c_x^k + \langle \tilde{g}_{\mathbf{c}}^k, x - x^k \rangle \quad (4.2.23)$$

to set up the cutting-plane models in (4.2.7), which gives

$$\check{f}^k(x) \leq f(x) + \epsilon_{\mathbf{f}} \text{ and } \check{c}^k(x) \leq c(x) + \epsilon_{\mathbf{c}} \text{ for all } k \text{ and all } x \in X. \quad (4.2.24)$$

Since both functions f and c in (4.2.2) are convex, the feasible set X is compact, and the oracle errors in (4.2.21) are bounded (it follows from (4.2.21) and (4.2.22) that $-\eta_{\mathbf{f}} \leq \epsilon_{\mathbf{f}}^x \leq \epsilon_{\mathbf{f}}$ and $-\eta_{\mathbf{c}} \leq \epsilon_{\mathbf{c}}^x \leq \epsilon_{\mathbf{c}}$), [114, Prop.XI.4.1.2] ensures that there exists a constant Λ such that

$$\Lambda \geq \max\{|\tilde{g}_{\mathbf{f}}^x|, |\tilde{g}_{\mathbf{c}}^x|\}; \text{ for all } x \in X. \quad (4.2.25)$$

Given a (approximate) lower bound f_{low}^k for the optimal value f_{\min} in (4.2.2), we define the inexact improvement function by

$$h_i^k(x) = \max\{f_x - f_{\text{low}}^k, c_x\}. \quad (4.2.26)$$

Due to inaccuracy from the oracle, we can have $h_i^k(x) < 0$.

Lemma 4.2.11. Consider the improvement function h_i^k given in (4.2.26). If there exists a sequence $\{x_{\text{rec}}^k\} \subset X$ such that $\lim_k h_i^k(x_{\text{rec}}^k) \leq 0$, then any cluster point of the sequence $\{x_{\text{rec}}^k\}$ is a η -solution to problem (4.2.2), with a possibly unknown error $\eta := \max\{\eta_{\mathbf{f}} + \epsilon_{\mathbf{f}}, \eta_{\mathbf{c}}\}$.

Proof. By (4.2.24) the lower bound updating at Step 5.2 ensures that $f_{\text{low}}^k \leq f_{\min} + \epsilon_{\mathbf{f}}$. If $\lim_k h_i^k(x_{\text{rec}}^k) \leq 0$, it follows by (4.2.26), the definition of η and oracle assumptions that

$$\begin{aligned} \text{(i)} \quad 0 &\geq \lim_k (f_{x_{\text{rec}}^k} - f_{\text{low}}^k) \geq \lim_k (f(x_{\text{rec}}^k) - \eta_{\mathbf{f}} - f_{\text{low}}^k) \geq \lim_k (f(x_{\text{rec}}^k) - \eta_{\mathbf{f}} - f_{\min} - \epsilon_{\mathbf{f}}), \\ \text{(ii)} \quad 0 &\geq \lim_k c_{x_{\text{rec}}^k} \geq \lim_k c(x_{\text{rec}}^k) - \eta_{\mathbf{c}}. \end{aligned}$$

Let \bar{x} be a cluster point of the sequence $\{x_{\text{rec}}^k\} \subset X$. Then $\bar{x} \in X$ and we conclude by (i) and (ii) that $f(\bar{x}) \leq f_{\min} + \eta_{\mathbf{f}} + \epsilon_{\mathbf{f}}$ and $c(\bar{x}) \leq \eta_{\mathbf{c}}$, i.e., \bar{x} is a η -solution to problem (4.2.2). \square

We now show that Lemma 4.2.6 still holds when all the three versions of Algorithm 4.2 are applied with the inexact improvement function h_i of (4.2.26).

Lemma 4.2.12. Let $l \geq 0$ be arbitrary, Λ be the bound in (4.2.25) and $\gamma \in (0, 1)$ be given. At iteration k of all versions (fc), (f) and (c) of Algorithm 4.2 applied to improvement function h_i of (4.2.26), assume that $h_i^k(x_{\text{rec}}^k) \geq 0$ holds. Then the following estimates are true:

$$\begin{aligned} |x^{k+1} - x^k| &\geq \frac{(1-\gamma)}{\Lambda} h_i^k(x_{\text{rec}}^k) \quad \text{if } k > k(l) \\ |x^{k+1} - \hat{x}^k| &\geq \frac{(1-\gamma)}{\Lambda} h_i^k(x_{\text{rec}}^k) \quad \text{if } k = k(l). \end{aligned}$$

Proof. In the following we focus on the version (fc), i.e., Algorithm 4.2(fc). The proofs for the remaining two versions can be obtained in a similar manner to the proof of Lemma 4.2.6.

Let k be arbitrary and $j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$ be given. By (4.2.9) combined with (4.2.10) and assumptions (4.2.21), we get

$$\begin{aligned} f_{x^j} + \langle \tilde{g}_{\mathbf{f}}^j, x^{k+1} - x^j \rangle &\leq f_{\text{lev}}^k \\ c_{x^j} + \langle \tilde{g}_{\mathbf{c}}^j, x^{k+1} - x^j \rangle &\leq 0. \end{aligned}$$

By applying the Cauchy-Schwarz inequality we thus derive

$$\begin{aligned} f_{x^j} - f_{\text{lev}}^k &\leq |\tilde{g}_{\mathbf{f}}^j| |x^{k+1} - x^j| \leq \Lambda |x^{k+1} - x^j| \\ c_{x^j} &\leq |\tilde{g}_{\mathbf{c}}^j| |x^{k+1} - x^j| \leq \Lambda |x^{k+1} - x^j|. \end{aligned}$$

Since $f_{\text{lev}}^k = f_{\text{low}}^k + \gamma h_i^k(x_{\text{rec}}^k)$ and $h_i^k(x_{\text{rec}}^k) \geq 0$ by assumption, we conclude that

$$\begin{aligned}
\Lambda |x^{k+1} - x^j| &\geq \max\{f_{x^j} - f_{\text{low}}^k - \gamma h_i^k(x_{\text{rec}}^k), c_{x^j}\} \\
&\geq \max\{f_{x^j} - f_{\text{low}}^k - \gamma h_i^k(x_{\text{rec}}^k), c_{x^j} - \gamma h_i^k(x_{\text{rec}}^k)\} \\
&= -\gamma h_i^k(x_{\text{rec}}^k) + \max\{f_{x^j} - f_{\text{low}}^k, c_{x^j}\} \\
&= -\gamma h_i^k(x_{\text{rec}}^k) + h_i^k(x^j) \\
&\geq (1 - \gamma) h_i^k(x_{\text{rec}}^k).
\end{aligned} \tag{4.2.27}$$

Assuming that $k > k(l)$, then the Oracle and Bundle Management steps assure that $k \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$. Hence the estimate (3.4.14) with $j = k$ provides the result. When $k = k(l)$, the stability center \hat{x}^k is chosen among the bundle members. Hence, there exists a $j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k$ such that $x^j = \hat{x}^k$ and (4.2.27) allows us to conclude this proof. \square

Similarly Lemma 4.2.7 can be derived by a slight change of conditions, as shown below.

Lemma 4.2.13. *Let $l \geq 0$ be arbitrary, D be the diameter of the feasible set X and Λ be the bound of (4.2.25). Consider the versions (fc), (f) and (c) of Algorithm 4.2, with improvement function h_i given in (4.2.26). Assume that at each bundle compression in all versions (fc), (f) and (c) of Algorithm 4.2 we include the aggregate indices in the bundle during the l -th cycle. Then, any iteration k in the l -th cycle (i.e., $k \in K^l$) may differ no more from $k(l)$ than the following bound, provided that $h_i^j(x_{\text{rec}}^j) > \delta_{\text{Tot}} > 0$ for all $j \leq k$:*

$$k - k(l) + 1 \leq \left(\frac{\Lambda D}{(1 - \gamma) h_i^k(x_{\text{rec}}^k)} \right)^2$$

The proof of the above lemma is identical to the proof of Lemma 4.2.7.

In what follows we make a distinction on the possible oracle types (4.2.21). We start by considering the most general case: upper oracles.

4.2.3.1 Inexact constrained level methods for upper oracles

An upper oracle is an oracle of type (4.2.21) wherein $\epsilon_{\mathbf{f}} > 0$ and/or $\epsilon_{\mathbf{c}} > 0$ in (4.2.22) might occur. This in particular implies that $\eta_{\mathbf{f}}^x, \eta_{\mathbf{c}}^x$ can be negative too and that the f - (c-) values can be overestimated.

Since in this case (4.2.24) holds, we can no longer keep Step 5 of Algorithm 4.2 as we can no longer assure that f_{lev}^k remains a lower bound for f_{min} . We thus suggest to change Step 5 as follows:

Step 5 (Upper) Try to solve the quadratic program (4.2.10). If no feasible solution is found:

Step 5.1 solve the linear problem

$$\text{val}_{LP} := \min \sum_{j \in J_{\mathbf{c}}^k} s_j \quad \text{s.t.} \quad x \in X, \quad \bar{c}^j(x) - s_j \leq 0 \quad \forall j \in J_{\mathbf{c}}^k. \tag{4.2.28}$$

If $\text{val}_{LP} > 0$, stop (the problem is $\epsilon_{\mathbf{c}}$ -infeasible); otherwise,

Step 5.2 Declare a critical iteration, set $l \leftarrow l + 1$, $k(l) \leftarrow k$, $f_{\text{low}}^k \leftarrow f_{\text{lev}}^k$ and choose $\hat{x}^k \in \{x^j : j \in J_{\mathbf{f}}^k \cap J_{\mathbf{c}}^k\}$. Return to Step 1.

If problem (4.2.10) is solved then move to Step 6.

We thus obtain the following convergence result.

Theorem 4.2.14. *Consider any version (fc), (f) or (c) of Algorithm 4.2, wherein Step 5 is replaced by Step 5 (Upper) applied with an upper oracle (4.2.21) and improvement function h_i^k given in (4.2.26). Suppose that $\delta_{\text{ToI}} = 0$ and that the algorithm does not terminate. Then $\lim_k h_i^k(x_{\text{rec}}^k) \leq 0$ and any cluster point of the sequence $\{x_{\text{rec}}^k\}$ is a η -solution to problem (4.2.2), with $\eta := \max\{\eta_{\mathbf{f}} + \epsilon_{\mathbf{f}}, \eta_{\mathbf{c}}\}$.*

Proof. Given the assumptions Lemma 4.2.13 holds. Proceed similarly as in the proof of Theorem 4.2.8 to show that $\lim_k h_i^k(x^k) \leq 0$, where f_{min} should be substituted by $\tilde{f} + \eta_{\mathbf{f}}$. Here \tilde{f} is defined as $\tilde{f} := \min_{x : c(x) \leq -\epsilon_c} f(x)$. It is easily observed that f_{low}^k is a proven lower bound for $\tilde{f} + \eta_{\mathbf{f}}$ and hence indeed $h_i^k(x_{\text{rec}}^k)$ must tend to zero. We can now use Lemma 4.2.11 to conclude. \square

Since the bounds given in Lemmas 4.2.7 and 4.2.13 are the same, we conclude that the complexity result of Theorem 4.2.9 also holds for inexact oracle satisfying (4.2.21). Up to our knowledge, general inexact oracles satisfying (4.2.21) have not been considered for (nonlinear) constrained level bundle methods. In [70] and [71] only lower oracles (see Section 4.2.3.3 below) are considered. The work [248] deals with upper oracles by applying a proximal bundle method.

Theorem 4.2.14 assumes that the algorithm does not terminate. This is not the case when problem (4.2.2) is detected to be ϵ_c -infeasible, i.e., $\{x \in X : \check{c}^k(x) \leq 0\} = \emptyset$ in Step 5.1. In the following section we deal with ϵ_c -infeasibility by making additional assumptions on problem (4.2.2) and oracle (4.2.21).

4.2.3.2 Inexact constrained level methods for upper oracles with on-demand accuracy

In this subsection we additionally assume that:

- problem (4.2.2) has a Slater point, i.e., there exists a point $x^s \in X$ such that $c(x^s) < 0$;
- demanding more accuracy on the c -information from (4.2.21) is possible (the upper bound ϵ_c in (4.2.22) can be decreased).

From (4.2.21) one directly derives that $\check{c}^k(x) \leq c(x) + \epsilon_c$ for all $x \in X$. This shows that upper oracles have nonetheless

$$\{x \in \mathbb{R}^n : c(x) \leq -\epsilon_c\} \subseteq \{x \in \mathbb{R}^n : \check{c}^k(x) \leq 0\} . \quad (4.2.29)$$

Now if the optimal value val_{LP} of the LP (4.2.28) is strictly positive, this means that the latter set is empty. Therefore, no proven feasible point could be produced by the algorithm and oracle anyway. Requesting that problem (4.2.2) admits a Slater point x^s

and that the oracle error is such that $c(x^s) < -\epsilon_c$ as in [248] would eliminate this case and the need for solving (4.2.28). However, how could we know if the inequality $c(x^s) < -\epsilon_c$ holds when x^s is unknown? In fact, as shown below we do not need to verify (at least explicitly) this inequality.

Let (\tilde{x}, \tilde{s}) be an optimal solution of problem (4.2.28). We say that \bar{c}^j is a wrong linearization if $\tilde{s}_j > 0$. That having been said, by assuming the existence of a Slater point $x^s \in X$ we may modify Step 5 (Upper) above in order to eliminate wrong linearizations $\bar{c}^j(\cdot)$ that cause ϵ_c -infeasibility. Eliminating indefinitely wrong linearizations from the bundle J_c^k may deteriorate the convergence. In order to avoid this problem, the adopted strategy is to clean the bundle J_c^k while requesting more accuracy from the c -oracle in (4.2.21). It follows from (4.2.29) that

$$\{x \in X : \bar{c}^k(x) \leq 0\} \neq \emptyset \text{ if } \epsilon_c \leq -c(x^s). \quad (4.2.30)$$

Hence, if the accuracy ϵ_c is decreased until it satisfies $\epsilon_c \leq -c(x^s)$, no wrong linearization will be identified and the algorithm will not stop before satisfying the stopping test. In what follows, we formalize the above idea:

Step 5 (Upper On-demand) Try to solve the quadratic program (4.2.10). If no feasible solution is found:

Step 5.1 Solve the linear problem (4.2.28) to get val_{LP} . If $val_{LP} > 0$ continue; otherwise go to Step 5.3.

Step 5.2 (Bundle Cleaning) Let (\tilde{x}, \tilde{s}) be a solution to (4.2.28). Set $J_c^k \leftarrow \{j \in J_c^k : \tilde{s}_j \leq 0\}$, $\epsilon_c \leftarrow \frac{\epsilon_c}{2}$ and go back to Step 5.

Step 5.3 declare a critical iteration, set $l \leftarrow l + 1$, $k(l) \leftarrow k$, $f_{low}^k \leftarrow f_{lev}^k$ and choose $\hat{x}^k \in \{x^j : j \in J_f^k \cap J_c^k\}$. Return to Step 1.

If problem (4.2.10) is solved then move to Step 6.

Every time Step 5.2 is accessed, the maximal c -oracle error is decreased by half. Hence, after finitely many applications of Step 5.2 the algorithm will determine $\epsilon_c \leq -c(x^s)$, and by (4.2.30) the existence of a Slater point ensures that no infinite loop between Steps 5 and 5.2 can occur. Therefore, Theorem 4.2.14 holds without any change. We emphasize that only the assumption of such Slater point $x^s \in X$ is enough for our strategy. In fact, we need to know neither x^s nor $c(x^s)$.

Since Step 5.2 requires more accuracy from the c -oracle by checking ϵ_c -infeasibility, the above approach can be considered as the on-demand accuracy type. Up to our knowledge, upper oracles with on-demand accuracy have not been studied so far in the literature of inexact bundle methods.

In what follows we no longer assume a Slater point. Furthermore, we focus on a particular case of oracle (4.2.21): lower oracles.

4.2.3.3 Inexact constrained level methods for lower oracles

A lower oracle is an oracle consistent with setting (4.2.21) allowing for one-sided errors only: in (4.2.22), we therefore assume $\epsilon_f \leq 0, \epsilon_c \leq 0$.

We thus derive $\eta_{\mathbf{f}}^x \geq -\epsilon_{\mathbf{f}}^x \geq 0$, $\eta_{\mathbf{c}}^x \geq -\epsilon_{\mathbf{c}}^x \geq 0$ for all $x \in X$. In particular, the cutting-plane models set up from approximate linearizations will remain below the true functions: using definition (4.2.7), the linearizations (4.2.23) provide for lower oracles cutting-plane models satisfying

$$\check{f}^k(x) \leq f(x) \quad \text{and} \quad \check{c}^k(x) \leq c(x) \quad \text{for all iteration counter } k \text{ and } x \in X.$$

In order to state our convergence result, we provide the oracle assumptions formally: for each given $x \in X$, an inexact oracle delivers

$$\text{the same information as oracle (4.2.21), however with } \epsilon_{\mathbf{f}}^x \leq 0, \epsilon_{\mathbf{c}}^x \leq 0. \quad (4.2.31)$$

As before, the upper bounds $\eta_{\mathbf{f}}$ and $\eta_{\mathbf{c}}$ in (4.2.22) can be unknown.

Lemma 4.2.15. *For lower oracles, the updating procedure of f_{low}^k at Step 5 of any version of Algorithm 4.2 assures $f_{\text{low}}^k \leq f_{\text{min}}$. Moreover, Lemma 4.2.11 holds with $\eta := \max\{\eta_{\mathbf{f}}, \eta_{\mathbf{c}}\}$.*

Proof. Lower oracles have the special property that

$$\{x \in \mathbb{R}^n : \check{f}^k(x) \leq f_{\text{low}}^k\} \cap \{x \in \mathbb{R}^n : c(x) \leq 0\} \subseteq \mathbb{X}^k,$$

due to the inequality $\check{c}^k(\cdot) \leq c(\cdot)$. If \mathbb{X}^k is found to be empty, we also have $f_{\text{low}}^k \leq \check{f}^k(x) \leq f(x)$ for all $x \in X$ with $c(x) \leq 0$, thus showing that f_{low}^k is indeed a proven lower bound for f_{min} . Moreover, since for lower oracles $\epsilon_{\mathbf{f}} = 0$, Lemma 4.2.11 holds with $\eta := \max\{\eta_{\mathbf{f}}, \eta_{\mathbf{c}}\}$. \square

We thus rely on Lemma 4.2.15 and Theorem 4.2.14 to give an asymptotic convergence result.

Theorem 4.2.16. *Consider any version (fc), (f) or (c) of Algorithm 4.2 applied with oracle (4.2.31) and improvement function h_i^k given in (4.2.26). Suppose that $\delta_{\text{Tot}} = 0$ and that the algorithm does not terminate. Then $\lim_k h_i^k(x_{\text{rec}}^k) \leq 0$ and any cluster point of the sequence $\{x_{\text{rec}}^k\}$ is a η -solution to problem (4.2.2), with $\eta := \max\{\eta_{\mathbf{f}}, \eta_{\mathbf{c}}\}$.*

Since lower oracles are likely to be less noisy than upper oracles, it is natural to expect that a solution obtained by any version of Algorithm 4.2 employing a lower oracle has a better quality. This is confirmed by comparing Theorems 4.2.14 and 4.2.16. Notice that the only difference between these two theorems is in the solution error η : $\max\{\eta_{\mathbf{f}} + \epsilon_{\mathbf{f}}, \eta_{\mathbf{c}}\}$ for upper oracles and $\max\{\eta_{\mathbf{f}}, \eta_{\mathbf{c}}\}$ for lower oracles.

If Algorithm 4.2 terminates at iteration k with $h_i^k(x_{\text{rec}}^k) \leq \delta_{\text{Tot}}$, then x_{rec}^k is a $(\max\{\eta_{\mathbf{f}}, \eta_{\mathbf{c}}\} + \delta_{\text{Tot}})$ -solution to problem (4.2.2). A more refined solution can be ensured if the oracle errors $\eta_{\mathbf{f}}^{x^k}$ and $\eta_{\mathbf{c}}^{x^k}$ vanish asymptotically for special iterates x^k . This case is addressed in the following section.

4.2.3.4 Constrained level bundle methods for lower oracles with on-demand accuracy

Following the lead of [71] we denote $x^k \in X$ a f -substantial iterate if the inexact value of the function f_x^k meets a descent target $f_{\text{tar}}^k \in \mathbb{R}$. Similarly, we denote $x^k \in X$ a c -substantial iterate if the inexact constraint value c_x^k meets a feasibility target $c_{\text{tar}}^k \in \mathbb{R}$. We call an iterate substantial if it is both f - and c -substantial. Moreover, we call substantial set \mathcal{S} , the index set gathering iterations that provide substantial iterates.

The aim of this section is to make a few changes in Algorithm 4.2 to deal with lower oracles that cast into the setting of (4.2.21), with errors that vanish for substantial iterates. Once again, $\epsilon_f \leq 0, \epsilon_c \leq 0$ in (4.2.22). We also assume that for each given point $x \in X$, upper bound $\eta \geq 0$ for the oracle error, and targets f_{tar} and c_{tar} , an inexact oracle delivers

$$\left\{ \begin{array}{l} \text{the same information as oracle (4.2.21), with:} \\ \epsilon_f^x \leq 0, \epsilon_c^x \leq 0 \\ \eta_f^x \leq \eta \text{ if } f_x \leq f_{\text{tar}} \\ \eta_c^x \leq \eta \text{ if } c_x \leq c_{\text{tar}}. \end{array} \right. \quad (4.2.32)$$

Notice that the upper bound for the oracle error η is given as input; therefore, it is known and controllable. Moreover, oracle (4.2.32) provides information with accuracy up to η for substantial iterates. We denote an oracle satisfying (4.2.32) by oracle with on-demand accuracy. This kind of oracles was introduced in [44] for unconstrained (or polyhedral constrained) convex nonsmooth optimization problems. The recent work [71] extends this concept to the constrained setting. Differently from the method in [71], our proposal allows for bundle compression, an important feature for practical applications. Moreover, in [71] the additional linear problem (4.2.17) is solved at each iteration k , while in Algorithm 4.2 it is optional.

We emphasize that oracle (4.2.32) is still quite general. Notice that for given $x^k = x$ and $\eta^k = \eta = 0$, the oracle provides exact information if both f_x^k and c_x^k meet the targets. This is a kind of partially exact oracle introduced in [130] for an unconstrained proximal bundle method, and further studied in [44] and [45]. Moreover, if both targets f_{tar}^k and c_{tar}^k are set as $+\infty$ for all iterations, then oracle (4.2.32) is an exact one. An asymptotically exact version of oracle (4.2.32) is obtained if, again, $f_{\text{tar}}^k = c_{\text{tar}}^k = \infty$ for all k and $\eta^k \rightarrow 0$. A combination between partially exact and asymptotically exact versions of oracle (4.2.32) is possible. For this setting, in order to get an exact solution to problem (4.2.2) the algorithm must force $\eta^k \rightarrow 0$ only for substantial iterates.

Let f_{low}^k be a proven lower bound for f_{min} at any iteration k . At any substantial iterate $j \in \mathcal{S}$, the oracle bound η^j is known and can be exploited in the definition of the improvement function. We therefore define:

$$h_{ae}^k(x^j) := \begin{cases} \max\{f_{x^j} - f_{\text{low}}^k, c_{x^j}\} + \eta^j & \text{if } j \in \mathcal{S} \\ h_{ae}^{j-1}(x_{\text{rec}}^{j-1}) & \text{otherwise.} \end{cases} \quad (4.2.33)$$

Since we force $j = 0 \in \mathcal{S}$, the above rule is well defined.

Lemma 4.2.17. Consider the improvement function h_{ae}^k given in (4.2.33), and \mathcal{S} the substantial index set. It follows that $h_{ae}^k(x^j) \geq h^k(x^j) \geq 0$ for all $x^j \in X$ and $j \in \mathcal{S}$, with h^k defined in (4.2.8). Besides, if there exists an index set $\mathcal{K} \subset \mathcal{S}$ such that $\{x_{\text{rec}}^k\}_{k \in \mathcal{K}} \subset X$ and $\lim_{k \in \mathcal{K}} h_{ae}^k(x_{\text{rec}}^k) = 0$, then any cluster point of the sequence $\{x_{\text{rec}}^k\}_{k \in \mathcal{K}}$ is a solution to problem (4.2.2).

Proof. Note that for each $j \in \mathcal{S}$, $h_{ae}^k(x^j) = \max\{f_{x^j} + \eta^j - f_{\text{low}}^k, c_{x^j} + \eta^j\} \geq \max\{f_{x^j} + \eta_{\mathbf{f}}^{x^j} - f_{\text{low}}^k, c_{x^j} + \eta_{\mathbf{c}}^{x^j}\}$. By the oracle assumptions, $h_{ae}^k(x^j) \geq \max\{f(x^j) - f_{\text{low}}^k, c(x^j)\} = h^k(x^j)$. Therefore, $\lim_{k \in \mathcal{K}} h_{ae}^k(x_{\text{rec}}^k) = 0$ implies that $\lim_{k \in \mathcal{K}} h^k(x_{\text{rec}}^k) = 0$. As function h^k is monotone, then $\lim_k h^k(x_{\text{rec}}^k) = 0$ and the stated result follows from Lemma 4.2.3. \square

With such a new setting, in order to show convergence of Algorithm 4.2 to an exact solution of problem (4.2.2) we need to force η^k to zero, for substantial iterates. This is done by controlling the oracle error at Step 5 of Algorithm 4.2 as follows:

Step 5 (On-demand) This step is similar as Step 5, however, if problem (4.2.10) is solved go to Step 5.1 before going to Step 6.

Step 5.1 (Noise Updating) Update the oracle error $\eta^{k+1} \leftarrow \theta h_{ae}^{k(l)}(x_{\text{rec}}^{k(l)})$, for a given $\theta \in (0, (1 - \gamma)^2)$.

Step 5.2 (Target Updating) Set $f_{\text{tar}}^{k+1} \leftarrow f_{\text{lev}}^k + (1 - \gamma)h_{ae}^k(x_{\text{rec}}^k)$ and $c_{\text{tar}}^{k+1} \leftarrow (1 - \gamma)h_{ae}^k(x_{\text{rec}}^k)$. Go to Step 6.

We now consider the following result based on Lemma 4.2.6.

Lemma 4.2.18. Let $\Lambda > 0$ given in (4.2.25) and parameters $\gamma \in (0, 1)$ and $\theta \in (0, (1 - \gamma)^2)$ in Algorithm 4.2, with Step 5 replaced by Step 5 (On-demand) and improvement function h^k replaced by h_{ae}^k given in (4.2.33). Then, at iteration k of any version (fc), (f) or (c) of Algorithm 4.2, the following estimates hold:

$$|x^{k+1} - x^k| \geq \frac{[(1 - \gamma)^2 - \theta]}{\Lambda} h_{ae}^k(x_{\text{rec}}^k) \text{ if } k \in K^l \text{ is such that } k > k(l).$$

Proof. We can proceed as in Lemma 4.2.12 to show that version (fc) satisfies

$$f_x^k - f_{\text{lev}}^k \leq |\tilde{g}_{\mathbf{f}}^k| |x^{k+1} - x^k| \leq \Lambda |x^{k+1} - x^k| \quad (4.2.34)$$

$$c_x^k \leq |\tilde{g}_{\mathbf{c}}^k| |x^{k+1} - x^k| \leq \Lambda |x^{k+1} - x^k|. \quad (4.2.35)$$

Suppose first that x^k is a substantial iterate: $k \in \mathcal{S}$. Since $f_{\text{lev}}^k = f_{\text{low}}^k + \gamma h_{ae}^k(x_{\text{rec}}^k)$ and $h_{ae}^k(x_{\text{rec}}^k) \geq 0$ by Lemma 4.2.17, we conclude that

$$\begin{aligned} \Lambda |x^{k+1} - x^k| &\geq \max\{f_x^k - f_{\text{low}}^k - \gamma h_{ae}^k(x_{\text{rec}}^k), c_x^k\} \\ &\geq \max\{f_x^k - f_{\text{low}}^k - \gamma h_{ae}^k(x_{\text{rec}}^k), c_x^k - \gamma h_{ae}^k(x_{\text{rec}}^k)\} \\ &= -\gamma h_{ae}^k(x_{\text{rec}}^k) + \max\{f_x^k - f_{\text{low}}^k, c_x^k\} \\ &= -\gamma h_{ae}^k(x_{\text{rec}}^k) + h_{ae}^k(x^k) - \eta^k, \end{aligned}$$

where the equality follows from (4.2.33), due to the assumption that $k \in \mathcal{S}$. Since $k \in K^l$ is such that $k > k(l)$, we have by Step 5 (On-demand) that $\eta^{k+1} = \eta^k$, and by Step 3

$h_{ae}^k(x_{\text{rec}}^k) > (1 - \gamma)h_{ae}^{k(l)}(x_{\text{rec}}^{k(l)})$. All together with the identity $\eta^{k+1} = \theta h_{ae}^{k(l)}(x_{\text{rec}}^{k(l)})$ by Step 5 (On-demand) we conclude that

$$\begin{aligned} \Lambda|x^{k+1} - x^k| &\geq (1 - \gamma)h_{ae}^k(x_{\text{rec}}^k) - \theta h_{ae}^{k(l)}(x_{\text{rec}}^{k(l)}) \\ &> (1 - \gamma)^2 h_{ae}^{k(l)}(x_{\text{rec}}^{k(l)}) - \theta h_{ae}^{k(l)}(x_{\text{rec}}^{k(l)}) \\ &\geq [(1 - \gamma)^2 - \theta] h_{ae}^{k(l)}(x_{\text{rec}}^{k(l)}) \\ &\geq [(1 - \gamma)^2 - \theta] h_{ae}^k(x_{\text{rec}}^k) \end{aligned}$$

where the last inequality is due to the monotonicity of $\{h_{ae}^k(x_{\text{rec}}^k)\}$. Hence, the result follows for $k \in \mathcal{S}$.

Under the assumption that $k \notin \mathcal{S}$, i.e., x^k is not a substantial iterate, we have (a) $f_x^k > f_{\text{tar}}^k$, and/or (b) $c_x^k > c_{\text{tar}}^k$.

– Case (a): the inequality $f_x^k > f_{\text{tar}}^k$ implies $f_x^k - f_{\text{lev}}^{k-1} > (1 - \gamma)h_{ae}^{k-1}(x_{\text{rec}}^{k-1})$. Since $k \in K^l$ is such that $k > k(l)$, we have that $f_{\text{lev}}^k \leq f_{\text{lev}}^{k-1}$. Therefore, it follows by monotonicity of h_{ae}^k that

$$f_x^k - f_{\text{lev}}^k \geq f_x^k - f_{\text{lev}}^{k-1} > (1 - \gamma)h_{ae}^{k-1}(x_{\text{rec}}^{k-1}) \geq (1 - \gamma)h_{ae}^k(x_{\text{rec}}^k) \geq [(1 - \gamma)^2 - \theta]h_{ae}^k(x_{\text{rec}}^k),$$

and the result follows from (4.2.34).

– Case (b): in this version, $c_x^k > (1 - \gamma)h_{ae}^{k-1}(x_{\text{rec}}^{k-1}) \geq (1 - \gamma)h_{ae}^k(x_{\text{rec}}^k) \geq [(1 - \gamma)^2 - \theta]h_{ae}^k(x_{\text{rec}}^k)$, and the result follows from (4.2.35).

Since the three considered possibilities ($k \in \mathcal{S}$, case (a) and case (b)) cover Algorithms 4.2(fc), 4.2(f) and 4.2(c), the stated result follows. \square

In what follows, D denotes again the diameter of compact set X .

Lemma 4.2.19. *In the setting of Lemma 4.2.18, consider the versions (fc), (f) and (c) of Algorithm 4.2, with improvement function h_{ae}^k given in (4.2.33) and Step 5 replaced by Step 5 (On-demand). Assume that at each bundle compression in all versions (fc), (f) and (c) of Algorithm 4.2 we include the aggregate indices in the bundle during the l -th cycle. Then, any iteration k in the l -th cycle (i.e., $k \in K^l$) may differ no more from $k(l)$ than the following bound, provided that $h_{ae}^j(x_{\text{rec}}^j) > \delta_{\text{Tot}} > 0$ for all $j \leq k$:*

$$k - k(l) + 1 \leq 1 + \left(\frac{\Lambda D}{[(1 - \gamma)^2 - \theta]h_{ae}^k(x_{\text{rec}}^k)} \right)^2$$

Proof. Since (4.2.16) holds, we can proceed as in Lemma 4.2.7 to conclude that

$$D^2 \geq |x^{k+1} - \hat{x}^k|^2 \geq |x^{k(l)+1} - \hat{x}^{k(l)}|^2 + \sum_{j=k(l)+1}^k |x^{j+1} - x^j|^2, \text{ for } k \in K^l, l \geq 0.$$

Together with Lemma 4.2.18, this yields the bounds

$$D^2 \geq \sum_{j=k(l)+1}^k \left(\frac{[(1 - \gamma)^2 - \theta]h_{ae}^k(x_{\text{rec}}^k)}{\Lambda} \right)^2 = \left(\frac{[(1 - \gamma)^2 - \theta]h_{ae}^k(x_{\text{rec}}^k)}{\Lambda} \right)^2 (k - k(l)),$$

and result follows. \square

The main convergence result is given in the following theorem.

Theorem 4.2.20. *Consider any version (fc), (f) or (c) of Algorithm 4.2 applied with oracle (4.2.32) and improvement function h_{ae}^k given in (4.2.33). Suppose that Step 5 is replaced by Step 5 (On-demand), and that parameters are chosen to satisfy $\gamma \in (0, 1)$ and $\theta \in (0, (1 - \gamma)^2)$. Suppose also that $\delta_{\text{Tot}} = 0$ and that the algorithm does not terminate. Then $\lim_k h_{ae}^k(x_{\text{rec}}^k) = 0$ and any cluster point of the sequence $\{x_{\text{rec}}^k\}$ is a solution to problem (4.2.2).*

Proof. If any version of Algorithm 4.2 stops at iteration k with $h_{ae}^k(x_{\text{rec}}^k) = 0$, then by (4.2.33) we have that $k \in \mathcal{S}$. Lemma 4.2.17 ensures that $(0 =) h_{ae}^k(x_{\text{rec}}^k) \geq h^k(x_{\text{rec}}^k) \geq 0$; thus x_{rec}^k is an optimal solution to problem (4.2.2).

If the algorithm does not stop, then by Lemma 4.2.19 (each cycle K^l is finite) we conclude that $l \rightarrow \infty$. Let \mathcal{W} (respec. \mathcal{L}) be an index set gathering counters l updated at Step 3 (respec. Step 5 (On-demand)) of (any version of) Algorithm 4.2.

Let l_i be the i -th index in $\mathcal{W} = \{l_1, l_2, \dots\}$. Therefore, the inequality $h_{ae}^{k(l_i)}(x_{\text{rec}}^{k(l_i)}) \leq (1 - \lambda)h_{ae}^{k(l_{i-1})}(x_{\text{rec}}^{k(l_{i-1})})$ holds even if there exists some $l \in \mathcal{L}$ such that $l_{i-1} < l < l_i$. A recursive application of the previous inequality gives us

$$0 \leq h_{ae}^{k(l_i)}(x_{\text{rec}}^{k(l_i)}) \leq (1 - \lambda)h_{ae}^{k(l_{i-1})}(x_{\text{rec}}^{k(l_{i-1})}) \leq \dots \leq (1 - \gamma)^i h_{ae}^{k(0)}(x_{\text{rec}}^{k(0)}).$$

As both function f and c are finite valued and the oracle error is bounded by $\eta^0 < \infty$, if \mathcal{W} has infinitely many indices then $\lim_i h_{ae}^{k(l_i)}(x_{\text{rec}}^{k(l_i)}) = 0$. Moreover, $\lim_k h_{ae}^k(x_{\text{rec}}^k) = 0$ by monotonicity; see (4.2.33).

Let us now suppose that \mathcal{W} is finite. As $l \rightarrow \infty$, then the index set \mathcal{L} has infinitely many indices. Since we are dealing with lower oracles, Lemma 4.2.15 shows that f_{low}^k is a proven lower bound for $f_{\text{min}} < \infty$. As $f_{\text{low}}^{k(l+1)}$ is increased at Step 5 (On-demand) of the algorithm by an amount of $\gamma h_{ae}^k(x_{\text{rec}}^k)$ for each $l \in \mathcal{L}$, we obtain that $\lim_k h_{ae}^k(x_{\text{rec}}^k) = 0$ (otherwise we would have $f_{\text{low}}^k \uparrow \infty$, which contradicts $f_{\text{low}}^k \leq f_{\text{min}} < \infty$).

We have shown in both cases (\mathcal{W} finite or infinite) that $\lim_k h_{ae}^k(x_{\text{rec}}^k) = 0$. Hence, it follows from (4.2.33) that there exists an index set $\mathcal{K} \subset \mathcal{S}$ such that $\lim_{k \in \mathcal{K}} h_{ae}^k(x_{\text{rec}}^k) = 0$, and together with Lemma 4.2.17 we have the stated result. \square

Theorem 4.2.20 ensures that if $\delta_{\text{Tot}} = 0$, then the algorithm eventually finds an exact solution to problem (4.2.2). This is not necessary the case if the algorithm is employed with upper oracles with on-demand accuracy; see Theorem 4.2.14 and comments in Section 4.2.3.2.

4.2.4 Numerical experiments

4.2.4.1 Benchmark Instance

For Benchmarking purposes we will investigate a Joint-Chance-Constrained Programming problem coming from cascaded reservoir management. We refer to section 4.1.5 for

an analysis of this problem and its adequacy with the structure exhibited in this section. We also refer to section 5.1 for a full description of the model. One could also consult [246].

We recall, once again, that these problems fit in the abstract structure (4.2.4), which we provide here with a few more details:

$$\begin{cases} \min & \langle q, x \rangle \\ \text{s.t.} & \mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x] \geq p \\ & \tilde{A}x \leq \tilde{b} \\ & x \geq 0. \end{cases} \quad (4.2.36)$$

In (4.2.36), $\xi \in \mathbb{R}^m$ is a Gaussian random vector with variance-covariance matrix Σ and zero mean (we have explicitly extracted the non-zero average in a^r, b^r). The Joint Chance Constraint arises since we wish to make sure that the volumes in the reservoirs remain within bounds with high enough probability. The latter volumes are impacted by random inflows and the turbining strategy. The latter will thus be rendered robust by (4.2.36). The polyhedral constraint set $\tilde{A}x \leq \tilde{b}$ results from modelling flow constraints, bounds on turbining, pumping and water valuation. Our first numerical experience consists of optimizing a hydro valley as in Figure 4.1(a). The dimension of the vector x in this instance is 672 and ξ has dimension 48. The key difficulty of problems of the type (4.2.36) is the dimension of ξ . The dimension of x can readily be increased without much impact on computation time.

In order to compute a (sub-)gradient of the chance constraint involved in (4.2.36), we use the formulae derived in Lemma 3.2.1. That formulae builds a link between computing a component of the partial derivative and the evaluation of probabilities of the form appearing in (4.2.36). Evaluating the latter probabilities requires using efficient numerical software such as Genz' Code [83, 84]. The latter code takes as an entry a precision ε^g , up to which the probabilities are evaluated. For instance picking $\varepsilon^g = 1e^{-4}$ implies that all probabilities are exact up to this precision. This precision allows us to control the “inexactness” of our Oracle.

A second instance of problem (4.2.36) consists of the Hydro valley appearing in Figure 4.1(b). We will consider 3 variants of this latter instance, wherein the dimension of the random vector ξ appearing in (4.2.36) varies from 48, 96 up to 168. The dimension of vector x is 888 in all variants of this instance. The first case occurs when random inflows are considered on reservoirs “COCHE” and “MONTRIGON”. The random vector has dimension 96 when reservoirs “BREVIERES” and “LA RAIE” are also impacted by uncertainty. Finally in the last case all reservoirs are assumed to have random inflows.

4.2.4.2 Preconditioning

Looking carefully at the quadratic program (QP) (4.2.10), one readily observes that the objective function has approximate order of magnitude $O(\frac{1}{2} \|x\|^2)$. Indeed, \hat{x}^k is typically proportional to the last iterate, so $\frac{1}{2} \|x - \hat{x}^k\|^2 = \frac{1}{2} x^\top x - (\hat{x}^k)^\top x + (\hat{x}^k)^\top \hat{x}^k \approx -\frac{1}{2} x^\top x$, by eliminating the constant. If x has large components, this may generate numerical

problems for the QP solver (even with CPLEX 12.2). We have therefore scaled the objective function by an appropriate constant in quite a similar way as one would do with a proximal bundle method. Returning to section 4.1 it appears that $\mu_0 = 1e^{-6}$ would make a nice choice.

A second problem might occur when the polyhedral system $\tilde{A}x \leq \tilde{b}$ has disproportionate components with respect to the constraint $\check{c}^k(x) \leq 0$. We have thus computed $b_{ratio} = \frac{\max_{i=1}^v |\tilde{b}_i|}{\min_{i=1}^v |\tilde{b}_i| + 1}$, where $\tilde{b} \in \mathbb{R}^v$. Let $g^\top x \leq e$ be an arbitrary constraint figuring in $\check{c}^k(x) \leq 0$. We have replaced it with $sg^\top x \leq se$, where $s = \frac{b_{ratio}}{\|g\|_\infty}$. This scaling also has to be used for the supporting hyperplane approach, shortly described below.

4.2.4.3 The various compared algorithms

An elementary method for solving problems of the type (4.2.36) is the supporting hyperplane method, which we will label Alg.[182]. We refer to section 2.8.4 for some historical comments on this method and to paragraph 4.1.6.1.3 for more specific information on this algorithm. The stopping criteria is based on obtaining a solution with proven optimality gap δ_{To1} . This algorithm requires a Slater point, which is computationally expensive to obtain.

We will also compare our Level Method with the Algorithm of section 4.1. We will refer to it as Alg.PB. Neither algorithm has used bundle compression. Since the considered inexact oracle is of the Upper type, the algorithm suggested in [71] is not suitable for our setting. This is why we have decided not to incorporate the algorithm of [71] in our numerical runs.

The numerical results show that Alg.LB:

1. Converges faster than Alg.PB to a solution with similarly optimality when parameters are appropriately set. This especially holds if the dimension is big. This might be due to the naive way in which the proximal parameter is updated. Nonetheless, even when parameters are not-fine tuned Alg.LB does equally well as Alg.PB.
2. Both the proximal and level bundle method outperform Alg.[182].

4.2.4.4 An extensive discussion of numerical results

It was shown in section 4.1 that an appropriate scaling of the constraints was needed in order to obtain correct numerical behaviour. This meant that instead of looking at the constraints $c(x) \leq 0$ we numerically deal with $Kc(x) \leq 0$ for an appropriately chosen constant $K > 0$. For the Algorithm of section 4.1 the constant K impacts both the constraints of the quadratic program and the definition of the improvement function. For our Algorithm 4.2 it turns out that scaling is only needed for the definition of the improvement function. It is then clear that $K > 0$ allows us to define an implicit

preference for improving first feasibility (large K values) or the objective function (small K values). This is also illustrated in Table 4.7.

Table 4.7: Comparison of Algorithms (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 5e^{-4}$. The number of iterations at which (4.2.10) is found infeasible is indicated in between brackets.

method	Obj. Value	Nb. Iter. [-(4.2.10)]	CPU time (mins)	parameters
Alg.[182]	-103363	16	7.2	$\delta_{\text{To1}} = 1e^{-2}$
Alg.[182]	-104077	49	25.5	$\delta_{\text{To1}} = 1e^{-3}$
Alg.[182]	-104156	99	43.9	$\delta_{\text{To1}} = 1e^{-4}$
Alg.PB	-104160	94	13.6	$K = 5e^4$
Alg.PB	-104160	91	12.3	$K = 1e^4$
Alg.PB	-104159	99	14.4	$K = 1e^5$
Alg.PB	*	*	-	$K = 1e^3$
Alg.LB	-104159	90 [2]	11.4	$K = 5e^4, \gamma = 0.9, \delta_{\text{To1}} = 5$
Alg.LB	-104158	94 [3]	12.3	$K = 1e^4, \gamma = 0.9, \delta_{\text{To1}} = 5$
Alg.LB	-104159	81 [2]	10.4	$K = 1e^5, \gamma = 0.9, \delta_{\text{To1}} = 5$
Alg.LB	-104160	58 [4]	6.5	$K = 1e^3, \gamma = 0.9, \delta_{\text{To1}} = 5$
Alg.LB	-104160	139 [1]	20.4	$K = 1e^6, \gamma = 0.9, \delta_{\text{To1}} = 5$
Alg.LB	*	*	-	$K = 1e^2, \gamma = 0.9, \delta_{\text{To1}} = 5$
Alg.LB	*	*	-	$K = 1e^1, \gamma = 0.9, \delta_{\text{To1}} = 5$

One can also observe that Alg.LB is less sensitive to K than Alg.PB, which is a strong advantage. Nonetheless too small values of K make the algorithm stop with infeasible solutions. This is understandable in the setting of problem (4.2.36) since the constraint, a chance constraint, has a rather small value compared to the objective function. Very small K values make this term negligible for the Algorithm. The observed “divergence” of Alg.PB with a small K value is of another order. The algorithm does not manage to produce solutions with improved feasibility at all. The proximal term then slowly converges to infinity and induces numerical problems in solving the quadratic program. If one directly compares the nominal solution with $K = 5e^4$ and compares CPU time with the one figuring in Table 4.4 (first line) it turns out that Alg.PB is found to be faster here. This is induced by the fact that Genz' code was recoded (in C++) and an approximate 30% speed up was obtained in that manner.

The effect of γ on the convergence is investigated in Table 4.8. One can observe that the behaviour of the Algorithm with $\gamma < 0.5$ and $\gamma > 0.5$ is slightly different, but Algorithm 4.2 is not very sensitive to this choice. The number of iterations increases when γ is small since we will detect empty level sets more often. These iterations are however not very costly as we do not call the oracle when an empty level set is detected. The number of iterations seems to be rather stable when $\gamma > 0.5$.

In the standard variant of Alg.LB the level set (4.2.9) is detected to be empty when the quadratic solver attempting to solve (4.2.10) produces an infeasible solution. The lower bound f_{low}^k is then updated. Alternatively the lower bound can be updated by solving the additional linear program (4.2.17) in Step 5 of Algorithm 4.2. Since in these numerical experiments there is no bundle compression, solving (4.2.17) at each iteration does not

Table 4.8: Effect of γ (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 5e^{-4}$. The number of iterations at which (4.2.10) is found infeasible is indicated in between brackets.

method	Obj. Value	Nb. Iter. $[\neg(4.2.10)]$	CPU time (mins)	parameters
Alg.LB	-104162	109 [36]	9.1	$K = 1e^3, \gamma = 0.100, \delta_{\text{Tot1}} = 5$
Alg.LB	-104162	91 [17]	9.2	$K = 1e^3, \gamma = 0.250, \delta_{\text{Tot1}} = 5$
Alg.LB	-104162	61 [9]	6.5	$K = 1e^3, \gamma = 0.500, \delta_{\text{Tot1}} = 5$
Alg.LB	-104162	60 [7]	7.1	$K = 1e^3, \gamma = 0.667, \delta_{\text{Tot1}} = 5$
Alg.LB	-104162	62 [6]	5.5	$K = 1e^3, \gamma = 0.800, \delta_{\text{Tot1}} = 5$
Alg.LB	-104160	58 [4]	6.5	$K = 1e^3, \gamma = 0.900, \delta_{\text{Tot1}} = 5$
Alg.LB	-104160	80 [5]	9.2	$K = 5e^4, \gamma = 0.100, \delta_{\text{Tot1}} = 5$
Alg.LB	-104159	70 [6]	8.4	$K = 5e^4, \gamma = 0.250, \delta_{\text{Tot1}} = 5$
Alg.LB	-104161	73 [4]	9.4	$K = 5e^4, \gamma = 0.500, \delta_{\text{Tot1}} = 5$
Alg.LB	-104159	73 [3]	9.6	$K = 5e^4, \gamma = 0.667, \delta_{\text{Tot1}} = 5$
Alg.LB	-104159	76 [2]	8.2	$K = 5e^4, \gamma = 0.800, \delta_{\text{Tot1}} = 5$
Alg.LB	-104159	90 [2]	11.4	$K = 5e^4, \gamma = 0.900, \delta_{\text{Tot1}} = 5$

impact the convergence analysis of the algorithm; see comment (f) after Algorithm 4.2. The effect of this choice is reported in Table 4.9. In the setting of problem (4.2.36) the c -oracle is costly to call and hence solving an additional linear program (or even quadratic program) has a negligible effect on computation time. A further effect of adding (4.2.17) to step 5 of the Algorithm is that (4.2.9) will never be empty anymore.

Table 4.9: Effect of adding (4.2.17) to Step 5. (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 5e^{-4}$.

method	Obj. Value	Nb. Iter.	CPU time (mins)	parameters
Alg.LB	-104161	35	4.4	$K = 1e^3, \gamma = 0.8, \delta_{\text{Tot1}} = 5$
Alg.LB	-104158	65	9.2	$K = 5e^4, \gamma = 0.8, \delta_{\text{Tot1}} = 5$

In Figure 4.3 we compare the iterative process of the first case in Table 4.9 with its counterpart in Table 4.8. Top graphics show the evolution of the sequences $\{f(x^k)\}$, $\{f_{\text{low}}^k\}$ and $\{f_{\text{lev}}^k\}$ along the iterations. The bottom left graphic compares the two sequences of $\{f_{\text{low}}^k\}$ corresponding to the following two cases: LP (4.2.17) is never solved, and (4.2.17) is solved at each iteration of Algorithm 4.2. Remember that when LP (4.2.17) is never solved, the lower bound f_{low}^k is updated only when the level set (4.2.9) is empty. We thus conclude by the stairs in the bottom right graphic in Figure 4.3 that the level set was detected to be empty six times in a total of sixty two iterations (as shown at line 5 of Table 4.8). The Bottom right graphic shows the sequences $\{c(x^k)\}$ for both cases. Notice that when solving LP (4.2.17) the resulting sequence $\{x^k\}$ is nearer to feasibility.

Up until now we have worked with an reasonably precise oracle: $\varepsilon^g = 5e^{-4}$. It is of interest to examine how results are when we are working with a precise Oracle: $\varepsilon^g = 1e^{-4}$. In table 4.3 it was shown that a precision $\varepsilon^g = 1e^{-5}$ led to unreasonable computation times. We will therefore not investigate this choice here. Table 4.10 provides these results.

When looking at the results of Table 4.10 one can observe that both Alg.LB and Alg.PB

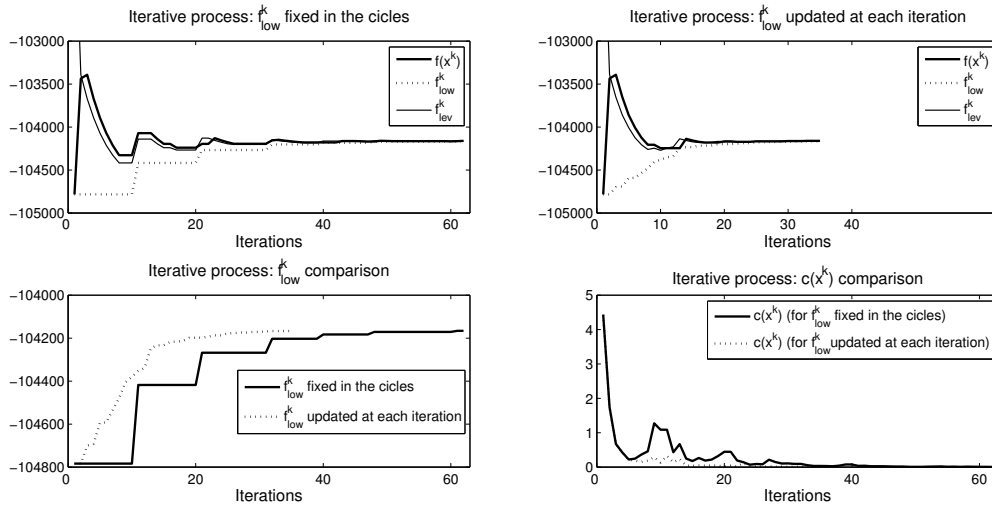


Figure 4.3: Comparison of Algorithm 4.2 when solving LP (4.2.17).

Table 4.10: Effect of precise Oracle. (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 1e^{-4}$.

method	Obj. Value	Nb. Iter.	CPU time (mins)	parameters
Alg.[182]	-104154	107	961.5	$\delta_{\text{To1}} = 1e^{-4}$
Alg.PB	-104160	89	71.2	$K = 5e4$
Alg.LB	-104162	33	24.2	$K = 1e^3, \gamma = 0.8, \delta_{\text{To1}} = 5$

outperform Alg.[182] in an important way. In particular we can note that a solution with similar optimality as Alg.[182] is produced by Alg.PB as early as the 67th iteration, which would correspond to an approximate computation time of around 53 minutes. The level method does not produce any intermediately feasible solutions and directly stops at optimality.

We now turn ourselves towards the instance consisting of the Isère Valley (Figure 4.1(b)). For these instances no Slater point was available so if we wish to use Alg.[182] we need to compute such a point. We have obtained this point by applying our Level Method with an increased p level (See (4.2.36) and (4.2.5)). We have taken $p = 0.95$ as a numeric value. Now if we use the objective function q of (4.2.36) too in such an algorithm, it is clear that we end up with a very strong Slater point. For instance, in the Isr96 instance below it turns out that the Slater point is only 0.3% suboptimal. Clearly the Alg.[182] algorithm will only need to do a few iterations to converge and obtain the optimal solution. Such a comparison would therefore be incorrect with respect to the other Algorithms, which would in fact boil down to a comparison between Alg.LB with hybrid stopping criteria and the other algorithms. Moreover in practice, such as in Unit-Commitment problems wherein cascaded reservoir management are a sub-problem, one would typically compute a Slater point against an early poor estimate of the true “price vector”. See section 5.2 for such a Unit-Commitment decomposition setting. Then the above advantage would go away. We have thus arbitrarily set $q = 0$ when computing the Slater point. This time is not integrated in the results obtained below.

Returning to the work of section 4.1, we can notice that two parameters (α and β) allow the user to have a control over the severity of the descent test and the noise test respectively. In this benchmark we make them vary to a setting which was found beneficial there. As Alg.LB is concerned, we will pick the best parameter settings of Table 4.7. The scaling parameter K was investigated in the work [249] and this led to the choice $K = 1e5$. Again such a choice can be argued to be intuitive since it is roughly half the order of magnitude of the objective function. It appears that Alg.LB works best with a choice of K roughly 10 times smaller. This is again confirmed in Table 4.11.

When making the comparison on instance Isr96, some comments are in order. One should note that Alg.PB with parameter $\alpha = 1$ produces a solution with similar quality as Alg.[182] in the 129th iteration, which corresponds roughly to 90 minutes of CPU time. Algorithm Alg.PB with parameter $\alpha = 0$ produces such a solution in the 167th iteration, which corresponds to 118 minutes of CPU time. So both version greatly outperform Alg.[182]. Finally Alg.LB in its best setting largely outperforms both Alg.PB and Alg.[182]. This is quite an encouraging result.

Now we turn our attention to the analysis of instance Isr168. When looking at Alg.PB, with $\alpha = 1$, a solution with similar optimality as Alg.[182] is produced in the 153th iteration, which would correspond to about 480 minutes of CPU time. A substantial advantage over Alg.[182]. The setting with $\alpha = 0$ produces such a solution in the 177th iteration, with about 553 minutes of CPU time. The level method Alg.LB with $K = 1e^4$ produces a solution of such quality as early as the 42nd iteration, but slightly infeasible. Still feasibility is easily restored when a Slater point is available. Globally both bundle

Table 4.11: Comparison of Alg.PB and Alg.LB on the Isère Valley. (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 5e^{-4}$. The number of iterations at which (4.2.10) is found infeasible is indicated in between brackets. The indication $[\neg(4.2.17)]$ means that auxiliary problem (4.2.17) is not solved.

Instance	method	Obj. Value	Nb. Iter. [$\neg(4.2.10)$]	CPU time (mins)	parameters
Isr48	Alg.[182]	-175031	35	10.5	$\delta_{To1} = 1e^{-4}$
Isr48	Alg.PB	-175043	115	17.5	$K = 1e^5, \mu_0 = 1e^{-5}, \alpha = 0, \beta = -1$
Isr48	Alg.PB	-175043	88	11.2	$K = 1e^5, \mu_0 = 1e^{-5}, \alpha = 1, \beta = -1$
Isr48	Alg.PB	-175042	69	8.3	$K = 1e^5, \mu_0 = 1e^{-6}, \alpha = 1, \beta = -1$
Isr48	Alg.LB	-175039	66	10.0	$K = 1e^5, \gamma = 0.8, \delta_{To1} = 5$
Isr48	Alg.LB	-175040	38	5.4	$K = 1e^4, \gamma = 0.8, \delta_{To1} = 5$
Isr48	Alg.LB	-175040	63 [3]	8.6	$K = 1e^5, \gamma = 0.8, \delta_{To1} = 5, [\neg(4.2.17)]$
Isr48	Alg.LB	-175037	38 [4]	5.2	$K = 1e^4, \gamma = 0.8, \delta_{To1} = 5, [\neg(4.2.17)]$
Isr96	Alg.[182]	-175708	143	217.4	$\delta_{To1} = 1e^{-4}$
Isr96	Alg.PB	-175714	214	152.3	$K = 1e^5, \mu_0 = 1e^{-5}, \alpha = 0, \beta = -1$
Isr96	Alg.PB	-175715	159	110.9	$K = 1e^5, \mu_0 = 1e^{-5}, \alpha = 1, \beta = -1$
Isr96	Alg.PB	-175715	177	123.5	$K = 1e^5, \mu_0 = 1e^{-6}, \alpha = 1, \beta = -1$
Isr96	Alg.LB	-175713	122	82.5	$K = 1e^5, \gamma = 0.8, \delta_{To1} = 5$
Isr96	Alg.LB	-175713	94	48.4	$K = 1e^4, \gamma = 0.8, \delta_{To1} = 5$
Isr96	Alg.LB	-175710	115 [3]	75.3	$K = 1e^5, \gamma = 0.8, \delta_{To1} = 5, [\neg(4.2.17)]$
Isr96	Alg.LB	-175697	76 [4]	44.3	$K = 1e^4, \gamma = 0.8, \delta_{To1} = 5, [\neg(4.2.17)]$
Isr168	Alg.[182]	-175222	190	1504.7	$\delta_{To1} = 1e^{-4}$
Isr168	Alg.PB	-175236	284	888.4	$K = 1e^5, \mu_0 = 1e^{-5}, \alpha = 0, \beta = -1$
Isr168	Alg.PB	-175237	219	687.4	$K = 1e^5, \mu_0 = 1e^{-5}, \alpha = 1, \beta = -1$
Isr168	Alg.PB	-175237	188	573.5	$K = 1e^5, \mu_0 = 1e^{-6}, \alpha = 1, \beta = -1$
Isr168	Alg.LB	-175235	161	529.6	$K = 1e^5, \gamma = 0.8, \delta_{To1} = 5$
Isr168	Alg.LB	-175232	110	352.3	$K = 1e^4, \gamma = 0.8, \delta_{To1} = 5$
Isr168	Alg.LB	-175235	165 [3]	423.2	$K = 1e^5, \gamma = 0.8, \delta_{To1} = 5, [\neg(4.2.17)]$
Isr168	Alg.LB	-175220	127 [5]	353.5	$K = 1e^4, \gamma = 0.8, \delta_{To1} = 5, [\neg(4.2.17)]$

methods have a significant advantage over Alg.[182]. The level method again offers advantages over the proximal variant since fine tuning the initial proximal parameter μ_0 is not very easy. A key difference between the proximal and the level bundle variant is that the former produces a feasible solution quickly and all successive stability centers are feasible solutions. The level method produces a feasible solution quite late, often near to the convergence.

As explained above the behaviour of Alg.[182] is heavily dependent on the choice of the Slater point. Surprisingly enough Alg.[182] performs quite well on instance Isr48, even though the obtained solution is sub-optimal. Furthermore, observing Table 4.10 and Table 4.7 it clearly appears that Alg.[182] suffers far more from increased oracle accuracy than Alg.PB or Alg.LB. Indeed the performance ratio moves from Alg.[182]/ Alg.PB = 3.2 to 13.5 and from 6.8 to 39.7 respectively. And this comparison does not integrate the loss of optimality incurred for Alg.[182]. This is confirmed by the results of Table 4.12.

Table 4.12: Effect of precise Oracle. (nne^x stands for $nn10^x$). Precision of Genz' code $\varepsilon^g = 1e^{-4}$.

Instance	method	Obj. Value	Nb. Iter.	CPU time (mins)	parameters
Isr48	Alg.[182]	-175037	39	58.5	$\delta_{\text{Tot1}} = 1e^{-4}$
Isr48	Alg.PB	-175041	81	35.1	$K = 1e5, \mu_0 = 1e^{-6}, \alpha = 1, \beta = -1$
Isr48	Alg.LB	-175039	39	16.2	$K = 1e^4, \gamma = 0.8, \delta_{\text{Tot1}} = 5$

4.2.4.5 Comments on Parameter Settings

The previous subsection was devoted to a comparison of Bundle Methods and the Supporting Hyperplane method for solving convex joint chance constrained programs. The latter method requires a Slater point, which might be tricky to obtain, but on the other hand the method has few parameters. This can be perceived as an advantage, since one might be afraid to have to spend much time in setting up appropriate choices. However this is untrue, since natural parameter choices for the Bundle methods led to good results. Indeed, both Bundle Methods have parameter settings to play with. The proximal variant, has choices for the definition of the nominal decrease, the noise test, the “magic” scaling parameter K and the initial proximal parameter μ_0 . The Level Method also has the scaling parameter K , γ and the stopping tolerance to set. We have however shown that results are relatively robust to changes in such parameters provided they are in the right zone. Also when moving from one instance to another (both coming from cascaded reservoir management) the same rules of thumb seem to give quite excellent initial choices. Globally for joint chance constraint programming we would suggest to

solve the following deterministic equivalent of (4.2.36):

$$\left\{ \begin{array}{l} \min \quad \langle q, x \rangle \\ \text{s.t.} \quad a^r + A^r x \leq 0 \\ \quad \quad 0 \leq b^r + A^r x \\ \quad \quad \tilde{A}x \leq \tilde{b} \\ \quad \quad x \geq 0 \end{array} \right. \quad (4.2.37)$$

and call x^d its optimal solution. Then $f_{\text{low}}^0 = \langle q, x^d \rangle$ is a good lower bound for (4.2.36), $K = \frac{1}{2} \langle q, x^d \rangle$ an appropriate choice for the magic Scaling parameter in the proximal Bundle Method. For the level method we should pick around $\frac{1}{10}$ th of that value. The initial proximal parameter should be set up according to what has been explained in section 4.2.4.2. According to our experience the quadratic term $\|x - \hat{x}^k\|^2$ should not exceed $1e^{11}$ and μ_0 can be set accordingly.

Chapter 5

Applications of JCCP in large scale Unit-Commitment through decomposition

In Energy Management, a key problem known as "unit-commitment" deals with finding a minimal cost production schedule. This schedule has to satisfy the operational constraints of each of the production units and meet customer load as closely as possible. Depending on the detail with which the operational constraints are modelled, many variants of this problem can be set up. Since operational constraints involve delays (start-up delays, etc...), the computed production schedule is often determined quite ahead of real-time. This allows for uncertainty to have a key impact on the "feasibility" and "optimality" of the executed production schedule. In practice, spinning reserves and intra-daily changes to the schedule allow the operator to partially account for uncertainty. Highly binding operational constraints might give rise to difficult situations, wherein the quest for "feasibility" induces a heavy cost. As such, computing a schedule having seen at least part of the plausible uncertainty, might turn out less costly eventually. This would be a key goal of Robust Unit-Commitment. However, unit-commitment problems are already very challenging in a deterministic setting. This is the consequence of the following facts:

1. each unit is subject to many complex technical constraints.
2. units exist in many varieties (thermal, hydraulic, contracts). Each type comes with a very specific set of constraints. Most of them requiring specific techniques for an efficient resolution.
3. the offer-demand equilibrium constraints couple all the units together

In order to tackle these large scale problems, the coupling constraints are often dualized, using Lagrangian techniques, leading to an effective price decomposition scheme ([38, 138, 79, 210]). Since the global unit-commitment problem is already challenging to solve in a deterministic setting due to its non-convex feasible sets and large scale, uncertainty is

often neglected, even though decisions are taken at least one day in advance. Uncertainty in unit-commitment problems comes at least from the following sources: customer load, renewable generation, inflows, unit availability. Integrating uncertainty in global unit-commitment will be quite challenging for the reasons outlined above. Hence, as a first necessary step, we will focus on hydro valley optimization. This will be investigated in section 5.1. In section 5.2 we will consider a global robust unit-commitment problem.

5.1 Robust Cascaded Reservoir Management

In the Lagrangian dualization setting of a unit-commitment problem, hydro valley optimization can be seen as a sub-problem. Alternatively, one can interpret this sub-problem as an optimization against market-prices. Complex dynamic constraints on watershed controls introduce combinatorial aspects in this sub-problem, making it difficult to solve. The focus of this section will therefore be on integrating uncertainty in hydro valley management.

The aforementioned combinatorial aspects result from formulating smoothness requests on watershed. From an engineering perspective it is undesirable to have turbining output increase and decrease rapidly over short time spans as this induces a strain on material. Other combinatorial elements can arise when modelling very realistic efficiency curves. We refer to [57] for an approach to deal (i.e., remove) the latter combinatorial elements. In hydro dominated systems, such as in Brazil, Scandinavia and Canada, the emphasis of accurate modelling lies on hydro generation and combinatorial optimization is common for cascaded reservoir management. We refer to [10, 75, 174, 162] for more details on such models. In thermal dominated systems, such as the French system, the modelling emphasis lies more on thermal generation. In these large-scale unit-commitment problems, such additional combinatorial elements are often neglected in order to have an acceptable computational burden (see [61]). We will make the same assumption. Integrating uncertainty and combinatorial elements in a cascaded reservoir model is quite challenging and will be investigated in future work. A potential entry point for such an approach would be the decomposition idea investigated in [75].

Uncertainty in cascaded reservoir management results from uncertainty on inflows and impacts the physical constraints of the system. Since decisions are taken prior to the observation of uncertainty, appropriate modelling approaches for integrating uncertainty have to be considered. The two main approaches are chance constrained programming and robust optimization. The main focus here is on the former, for the latter we refer to [13, 7, 3]. In dynamic decision processes, i.e., when decisions in later time periods are allowed to adapt to earlier observed uncertainty, the main approaches are Stochastic Dynamic Programming and SDDP (see [169]). Often the convenient hypothesis is made that uncertainty within the transition problem is known. This essentially makes the transition problem a deterministic problem. The latter choice is especially questionable when the transition problem covers a time span of a week such as assumed in [172].

Introduced by [32], probability constraints are quite an appealing tool for dealing with

uncertainty. In particular, when uncertainty arises in physical constraints, since they also offer a simple interpretation. Since their first introduction, chance constraints have become quite common in hydro valley management ([143, 63, 65, 142, 155, 265, 264, 244]), but often individual chance constraints are used and not joint chance constraints. Though a very appealing approximation, individual chance constraints unfortunately do not offer sufficient robustness (see [244]). Hydro reservoir models with joint chance constraints have been considered, for instance, in [185, 186, 244]. In [1] even a dynamic approach has been developed in this context. However, these models were comparatively simple from their structure (no serially linked reservoirs, no delay time between reservoirs, no realistic water value condition, no time series modelling of statistical data, small dimension). The main focus of [244] is on deriving an efficient gradient formula for joint chance constraints of a specific form. The latter form arises naturally in hydro reservoir management. The interest of the formula is then illustrated on a stylized hydro reservoir optimization problem.

This section is organized as follows. In section 5.1.1, we present our model for hydro reservoir management, where combinatorial constraints are neglected and random inflows are introduced. We give a detailed description of a real hydro valley, and present the main optimization problem. As the uncertainty on inflows is concerned, many statistical models are based on a deterministic trend (potentially dependent on explanatory variables) and a causal noise process. Since convexity results exist for specific classes of randomness and in particular Gaussian ones, it seems tempting to place ourselves in such a setting. Restricting uncertainty laws to such a setting, might seem restrictive at first. However, we will show that a large class of models is available, i.e., the class of causal time series models with Gaussian innovations ([217]).

In section 5.1.2, we present several alternatives to the joint chance constraint problem. In particular, we present an individual chance constraint problem and a robust model. We report results obtained when solving these various models on a realistic instance of a hydro valley management problem in section 5.1.3. The interest of joint chance constrained programming is illustrated by comparing results obtained on this hydro valley with those obtained from the alternative models.

5.1.1 Problem Description

In this section we will give a description of the hydro reservoir management problem. We will consider a discretized time horizon. To this end let $\tau = \{1, \dots, T\}$ denote the set of (homogeneous) time steps, where T denotes the last time step. Let Δt be the time step size expressed in hours. We will begin by providing problem constraints and the objective function. We will conclude with a paragraph highlighting the structure of the problem.

5.1.1.1 Problem Constraints

5.1.1.1.1 Topology A hydro valley can be seen as a set of connected reservoirs and associated turbines. We can therefore represent this with a directed graph. Let \mathcal{N} be the set of nodes and let A (of size $|\mathcal{N}| \times |\mathcal{N}|$) be the connection matrix, i.e., $A_{n,m} = 1$ whenever water released from reservoir n will flow into reservoir m . We will assume that D is the flow duration vector, i.e., D_m is the amount of time (measured in time steps) it takes for water to flow from upper reservoir m to its unique child. It is assumed that pumping is (nearly) instantaneous. Let $\mathcal{T} := \{g^i, i = 1, \dots, N_{\mathcal{T}}\}$ denote the set of turbines and $\mathcal{P} := \{p^i, i = 1, \dots, N_{\mathcal{P}}\}$ denote the set of pumping stations. We furthermore introduce the mapping $\sigma_{\mathcal{T}} : \{1, \dots, N_{\mathcal{T}}\} \rightarrow \mathcal{N}$ ($\sigma_{\mathcal{P}} : \{1, \dots, N_{\mathcal{P}}\} \rightarrow \mathcal{N}$) attributing to each turbine (pumping station) the reservoir number to which it belongs. We will also introduce the sets $\mathcal{A}(n) = \{m \in \mathcal{N} : A_{m,n} = 1\}$ and $\mathcal{F}(n) = \{m \in \mathcal{N} : A_{n,m} = 1\}$. The set $\mathcal{A}(n)$ is empty for uphill reservoirs and the set $\mathcal{F}(n)$ for downhill reservoirs. To each reservoir $n \in \mathcal{N}$ and for each time step $t \in \tau$ we associate its volume $V^n(t)$ in cubic hectometers hm^3 . The initial volume of each reservoir $n \in \mathcal{N}$ is denoted by $V^n(0)$, lower and upper bounds are $V_{min}^n(t)$ and $V_{max}^n(t)$ respectively.

5.1.1.1.2 Controls We will assume that each turbine (and pumping station) can be controlled for each time step. To this end we introduce the variables $x^i(t)$ for each $t \in \tau$ and $i = 1, \dots, N_{\mathcal{T}}$. In a similar way we introduce the variables $y^i(t)$ for the pumping stations. The units are in cubic meters per hour, i.e., m^3/h . Furthermore we assume that each of these variables are bounded:

$$\begin{aligned} 0 &\leq x^i(t) \leq \bar{x}^i(t), \quad \forall t \in \tau, i = 1, \dots, N_{\mathcal{T}} \\ 0 &\leq y^i(t) \leq \bar{y}^i(t), \quad \forall t \in \tau, i = 1, \dots, N_{\mathcal{P}}. \end{aligned} \tag{5.1.1}$$

5.1.1.1.3 Random Inflows We will assume that inflows (in m^3/h) in reservoirs are the result of some stochastic process. Let $A^n(t)$ denote this stochastic process for reservoir n . Not all reservoirs will have stochastic inflows, some of them will have deterministic inflows. This can be explained by the fact that top reservoirs have random inflows due to the melting of snow in the high mountains, whereas rain can be neglected for lower reservoirs. Let $\mathcal{N}^r \subseteq \mathcal{N}$ denote the set of reservoirs receiving random inflows. We will assume that the stochastic inflow process is the sum of a deterministic trend s_t^n and a causal process ([217]) generated by Gaussian innovations. To this end, let $\zeta^n(t)$ be a Gaussian white noise process, where $(\zeta^{k_1}(t), \dots, \zeta^{k_l}(t))$ is a Gaussian random vector of zero average and variance-covariance matrix $\Sigma(t)$ ($\{k_1, \dots, k_l\} = \mathcal{N}^r$). This last assumption means that the innovations for several reservoirs are correlated. This correlation alone will bring forth the correlation structure on inflows across reservoirs. This assumption is quite similar to the one made in mathematical finance wherein Brownian motions are correlated and thus generate correlated commodities.

We will assume independence between time steps of the ζ vector. Since $A^n(t)$ is a causal

process, we can write it as follows

$$A^n(t) = s_t^n + \sum_{j=0}^{\infty} \psi_j^n \zeta^n(t-j) = s_t^n + \sum_{j=t}^{\infty} \psi_j^n \zeta^n(t-j) + \sum_{j=0}^{t-1} \psi_j^n \zeta^n(t-j), \forall n \in \mathcal{N}^r, t \in \tau$$

for some coefficient vector ψ^n and infinite past before $t = 0$ (the beginning of the optimization horizon). We will assume that randomness before (and including) $t = 0$ is known and as such we can assume w.l.o.g. that the random inflow process can be written as

$$A^n(t) = s_t^n + \sum_{j=0}^{t-1} \psi_j^n \zeta^n(t-j), \forall n \in \mathcal{N}^r, t \in \tau. \quad (5.1.2)$$

For reservoirs $n \in \mathcal{N} \setminus \mathcal{N}^r$, we simply have $A^n(t) = s_t^n$.

5.1.1.1.4 Flow constraints and Volume bounds Each reservoir is subject to flow constraints induced by pumping and turbining. The following balance constraint applies

$$\begin{aligned} V^n(t) &= V^n(t-1) + \sum_{m \in \mathcal{A}(n)} \sum_{i \in \sigma_{\mathcal{T}}^{-1}[m]} x^i(t - D_m) \Delta t - \sum_{i \in \sigma_{\mathcal{T}}^{-1}[n]} x^i(t) \Delta t \\ &+ \sum_{m \in \mathcal{F}(n)} \sum_{i \in \sigma_{\mathcal{P}}^{-1}[m]} y^i(t) \Delta t - \sum_{i \in \sigma_{\mathcal{P}}^{-1}[n]} y^i(t) \Delta t \\ &+ s_t^n \Delta t + \sum_{j=0}^{t-1} \psi_j^n \zeta^n(t-j) \Delta t, \forall t \in \tau, n \in \mathcal{N}. \end{aligned} \quad (5.1.3)$$

The above equation is entirely deterministic except for the reservoirs $n \in \mathcal{N}^r$. In order to deal with this randomness and reservoir bounds we will therefore add the following constraints

$$\mathbb{P}[V_{min}^n(t) \leq V^n(t) \leq V_{max}^n(t) \forall t \in \tau, n \in \mathcal{N}^r] \geq p \quad (5.1.4)$$

$$V_{min}^n(t) \leq V^n(t) \leq V_{max}^n(t) \forall t \in \tau, n \in \mathcal{N} \setminus \mathcal{N}^r, \quad (5.1.5)$$

where \mathbb{P} is a probability measure and p a security level. Constraint (5.1.4) is a joint chance constraint. This means that we wish to satisfy all linear inequalities of the stochastic system simultaneously with high enough probability. This can be compared to a model with individual chance constraints, which is a model wherein we wish to satisfy each inequality with high enough probability, but taken separately. We will show (see section 5.1.3) that the latter model offers insufficient robustness.

5.1.1.1.5 Water Values In short term optimization problems (with time horizons ranging from several days up to a month) water values provide a way to associate a cost with used water. Incorporating no such cost in a short term optimization problem would inevitably lead to a maximum use of water on this specific time horizon, whereas water might be needed in later time periods. Water might be used to reduce the use of costly

thermal generation or as a security to avoid "black-outs" in difficult situations. Water values are obtained as the by product of (stochastic) dynamic programming approaches in mid term (time horizons ranging from 1 to 5 years).

In full generality water values depend on time, a multivariate random vector, the current water levels in all reservoirs and other quantities that can be considered as inventories or stocks (such as customer interruption options (see [264] for more details), i.e., an inventory globally very similar to the number of remaining exercise rights in swing options). As the effect of uncertainty is concerned, it is often averaged out on a set of reasonable scenarios in order to integrate unconditional water values in short term optimization. The stochastic dynamic programming algorithms typically deal with uncertainty effects rarely integrated in short term optimization such as stochastic fuel prices.

The multivariate stock dependency is only known approximately, if at all, since one quickly hits the curse of dimensionality of dynamic programming. In such cases, approaches such as approximate dynamic programming (ADP) ([43]), approximate dual dynamic programming (ADDP) ([86]), SDDP ([169, 173]) or aggregation approaches ([230, 229]) are applied in order to approximately solve the dynamic programming problem. In the ADP approach, it is commonly assumed that the continuation function of dynamic programming decomposes as a sum of 1 dimensional functions. Each function depending on a unique stock only. This then automatically results in single stock dependent water values. Even if water values would be available as multivariate functions, they would only be known on a set of grid points. If this is to be incorporated in short term optimization one surely needs interpolation techniques very similar to those explained in [40]. This interpolation approach leads to the introduction of binary variables in the optimization problem. Since multivariate effects in water values are only rarely known and integrating them induces combinatorial aspects, we will focus on single stock dependent water values.

As the temporal dependency is concerned it is often daily or intra-daily. Due to the average effect of climate on unit-commitment, some specific weeks are far more costly than surrounding weeks. Such weeks have peaking customer load and high risk of black outs. Such effects get reflected in the water values as well. These effects are moreover strengthened by averaging out stochastically dependent water values as explained above.

If we wish to incorporate water values in short term optimization, the latter temporal effect can either be neglected or taken into account. In the first case, we would value the differential between the end and the initial volume of a reservoir against water values at that time step. In the second approach we would either value volumes against water values at each time step or value local volumetric differences. The first approach would consider indifferently any two storage paths leading to the same end volume. When the short term time horizon is close to a month and one of the above difficult weeks is within this time horizon, from an operational view point two paths leading to the same end volume are not necessarily considered equivalent. It is therefore of interest to integrate the temporal dependence in order to reflect this feature. A second reason for integrating this effect is to provide a model that fits better with current practice. In practice, in order to control the storage path, a selection of time steps $\tau^s \subseteq \tau$ is made where artificially we

force $V_{min}^n(t) \approx V_{max}^n(t)$ for $t \in \tau^s$. Integrating the temporal dependency of water values in short term optimization is a natural way to have control over the storage path without risking to have an empty feasible set.

In this section we present a model for incorporating water values without reflecting temporal dependencies as the focus of the section is on Chance Constrained programming for hydro reservoir management. Upon valuing the volume at each time step against local water values, the presented model allows for a straightforward extension for incorporating the above discussed temporal effect of water values. We also discuss how valuing volumetric differences leads to a non-convex model.

5.1.1.1.5.1 Volume dependent water values Our aim is to set up a model which evaluates the expected amount of water in the reservoir at the end of the optimization horizon¹. This is necessary in order not to carry out the optimization at the expense of later periods of time. A possible way to do so is to subdivide the levels of each reservoir into a finite number of values from bottom to top as follows:

$$V_0^n, \dots, V_{K_n}^n \quad \forall n \in \mathcal{N}.$$

Each compartment $[V_{i-1}^n, V_i^n)$ is assigned a water value W_i^n (in $\text{€}/\text{m}^3$) such that

$$W_{i-1}^n > W_i^n \geq 0 \quad \forall n \in \mathcal{N} \quad \forall i = 1, \dots, K_n. \quad (5.1.6)$$

The value of the expected final water level $\mathbb{E}(V^n(T))$ of reservoir n is then simply the cumulative value of water in the compartments below:

$$\sum_{i \leq i^*} W_i^n (V_i^n - V_{i-1}^n) + W_{i^*}^n (\mathbb{E}(V^n(T)) - V_{i^*}^n), \quad i^* := \max\{i \mid \mathbb{E}(V^n(T)) \geq V_i^n\}.$$

Note that this value is an increasing function of the expected final level $\mathbb{E}V^n(T)$ despite the fact that water values are strictly decreasing from bottom to top.

Now, in order to avoid combinatorial arguments concerning the index i^* , we introduce auxiliary variables z_i^n indicating for each reservoir n the amount of water in compartment $[V_{i-1}^n, V_i^n)$. Of course, since all compartments have to be completely filled up to i^* , one has that

$$z_i^n = \begin{cases} V_i^n - V_{i-1}^n & i = 1, \dots, i^* \\ \mathbb{E}(V^n(T)) - V_{i^*}^n & i = i^* + 1 \\ 0 & i = i^* + 2, \dots, K_n \end{cases} \quad \forall n. \quad (5.1.7)$$

Then, the value of the final water level in reservoir n equals

$$\sum_{i=1}^{K_n} W_i^n z_i^n \quad \forall n. \quad (5.1.8)$$

¹In practice, one would evaluate the difference of the final and initial volume. The latter adds a constant to the objective function and can theoretically be omitted. In practice, it may generate some numerical difficulties, especially when large volumes are valued and turbinning/pumping capacity is small compared to the volume. In that case, relative changes in valuation induced by the controls are easily considered negligible. Moreover, the constant can easily be added.

We claim that the relations (5.1.7) for variables z_i^n can be replaced by the following relations in which the crucial index i^* is absent:

$$\sum_{i=1}^{K_n} z_i^n = \mathbb{E}(V^n(T)) - V_0^n \quad \forall n \quad (5.1.9)$$

$$0 \leq z_i^n \leq V_i^n - V_{i-1}^n \quad \forall n \forall i = 1, \dots, K_n. \quad (5.1.10)$$

The argument is as follows: as part of the overall objective function in our problem, we shall maximize the value of the final water level (5.1.8). Given the strictly decreasing order of water levels in (5.1.6) (from bottom to top), it is clear from (5.1.9) that the upper inequality in (5.1.10) will be satisfied as an equality as long as possible and that only the most upper compartment may not be completely filled. This of course is equivalent with (5.1.7) but avoiding the explicit description of that most upper compartment.

Since the initial volume $V^n(0)$ is known in advance, one can define variables $z_{0,i}^n$ in a similar way as z_i^n . It then follows that

$$\sum_{i=1}^{K_n} W_i^n (z_{0,i}^n - z_i^n) \quad (5.1.11)$$

is the cost of used water for reservoir $n \in \mathcal{N}$. The valuation induced by $\sum_{i=1}^{K_n} W_i^n z_{0,i}^n$ is in fact a constant and can be omitted.

5.1.1.1.5.2 Time and Volume dependent water values : Volumetric differences Our aim is to set up a model in which (expected) volumetric difference of adjacent time steps are valued with a water value that depends on time and volume. We will show that such a model is non-convex. We therefore suggest to extend the model of paragraph 5.1.1.1.5.1 instead. We begin by recalling the subdivision of the levels of each reservoir in (5.1.1.1.5.1). To each of these compartments we assign a water value $W_i(t)^n$ (in €/m³), $i = 1, \dots, K_n$, $n \in \mathcal{N}$. We assume that these are decreasing:

$$W_{i-1}^n(t) > W_i^n(t) \geq 0 \quad \forall i = 1, \dots, K_n, n \in \mathcal{N}, t \in \tau. \quad (5.1.12)$$

Assuming for simplicity of reasoning that the dependence on the compartments is absent. Then we wish to value

$$\sum_{n \in \mathcal{N}} \sum_{t \in \tau} W^n(t) \mathbb{E}(V^n(t) - V^n(t-1)). \quad (5.1.13)$$

For a fixed reservoir $n \in \mathcal{N}$ we define $i_{n,t}^* = \max \{i \mid \mathbb{E}(V^n(t)) \geq V_i^n\}$. Let $i = 1, \dots, K_n$ moreover be arbitrary and define $z_{x,i}^n(t)$ as negative variations of volume with respect to expected volume $\mathbb{E}(V^n(t))$ (i.e., turbinning). Similarly, we define $z_{y,i}^n(t)$ as the positive variations (i.e., pumping). Both variations are restricted to the interval $[V_{i-1}^n, V_i^n]$. This means that:

$$V^n(t+1) = V^n(t) + \sum_{i=1}^{K_n} (z_{y,i}^n(t) - z_{x,i}^n(t)).$$

Since $z_{x,i}^n(t)$ is defined as the negative variations with respect to the expected volume $\mathbb{E}(V^n(t))$, one has $z_{x,i}^n(t) = 0$ for all $i > i_{n,t}^* + 1$. By similar arguments we readily observe that $z_{y,i}^n(t) = 0$ for all $i < i_{n,t}^* - 1$. Moreover, one has:

$$z_{x,j}^n(t) > 0 \Rightarrow z_{x,i}^n(t) = \min \{V_i^n, \mathbb{E}(V^n(t))\} - V_{i-1}^n, i = j + 1, \dots, i_{n,t}^* + 1 \quad (5.1.14)$$

$$z_{y,j}^n(t) > 0 \Rightarrow z_{y,i}^n(t) = V_i^n - \max \{V_{i-1}^n, \mathbb{E}(V^n(t))\}, i = i_{n,t}^* - 1, \dots, j - 1. \quad (5.1.15)$$

It therefore follows that the appropriate generalization of (5.1.13) accounting for the compartments is:

$$\sum_{n \in \mathcal{N}} \sum_{t \in \tau} \sum_{i=1}^{K_n} W_i^n(t) (z_{x,i}^n(t) - z_{y,i}^n(t)). \quad (5.1.16)$$

The following constraints:

$$0 \leq z_{x,i}^n(t) \leq \max \{ \mathbb{E}(V^n(t)) - V_{i-1}^n, 0 \}, i = 1, \dots, K_n, n \in \mathcal{N} \quad (5.1.17)$$

$$0 \leq z_{y,i}^n(t) \leq \max \{ V_i^n - \mathbb{E}(V^n(t)), 0 \}, i = 1, \dots, K_n, n \in \mathcal{N} \quad (5.1.18)$$

assure that $z_{x,i}^n(t) = 0$ for all $i > i_{n,t}^* + 1$ and $z_{y,i}^n(t) = 0$ for all $i < i_{n,t}^* - 1$. One can also see that the requirements (5.1.14) and (5.1.15) are automatically satisfied by hypothesis (5.1.12) and objective function (5.1.21). The above non-linear constraints (5.1.17) and (5.1.18) can be transformed into quadratic constraints by applying Lemma 5.1.1. The same lemma then allows us to conclude the non-convexity of the model involving valuation (5.1.16).

Lemma 5.1.1. *Let $f, g, h : \mathbb{R}^n \rightarrow \mathbb{R}$ be finite-valued mappings. Then the following problems are equivalent:*

$$\min_{x \in \mathbb{R}^n} \{ f(x) : g(x) \leq [h(x)]^+ \} \quad (5.1.19)$$

and

$$\min_{x \in \mathbb{R}^n, \lambda \geq 0} \{ f(x) : (g(x) - h(x) + \lambda)g(x) \leq 0 \} \quad (5.1.20)$$

If moreover, both g and h are linear, the complementarity constraint appearing in (5.1.20) is quadratic. In the particular case, when g takes the form $g(x, z) = z$ and $h(x, z) = a^\top x$, $x \in \mathbb{R}^n, z \in \mathbb{R}$, the matrix defining the quadratic form of the inequality is given by

$$Q = \begin{pmatrix} 0 & -0.5a & 0 \\ -0.5a^\top & 1 & 0.5 \\ 0 & 0.5 & 0 \end{pmatrix}$$

and has n zero eigenvalues and 2 non-zero ones, $0.5 \pm 0.5\sqrt{\|a\|^2 + 2}$. The resulting quadratic constraint is non-convex.

Proof. For each feasible x for problem (5.1.19), one can find some $\lambda \geq 0$ such that (x, λ) is feasible for (5.1.20). This can be seen by case distinction on the sign of $h(x), g(x)$. By case distinction on the sign of $g(x)$ one can see that each feasible (x, λ) for problem

(5.1.20), yields a feasible x for problem (5.1.19). It is easily seen that the quadratic constraint is $y^\top Q y \leq 0$, with $y := (x, z, \lambda)$. Clearly any vector $(b, 0, 0)$ with $b^\top a = 0$ is an eigenvector with zero eigenvalue. This means that Q has at least $n - 1$ zero eigenvalues. Picking $v = (x, z, \lambda)$, equating $Qv = \mu v$, gives that $x = -\frac{1}{2\mu}z a$, $\lambda = \frac{1}{2\mu}z$ and shows that non zero eigenvalues μ must satisfy the equation $\mu^2 - \mu - (\frac{1}{4}a^\top a + \frac{1}{4}) = 0$. Solving this equation yields the two non zero eigenvalues summing up to one as given in the lemma. This implies that we have identified $n + 1$ out of $n + 2$ eigenvalues. Since $\text{Tr}(Q) = 1$, the last eigenvalue has to be zero too. The non-convexity follows since one eigenvalue of Q is negative. \square

5.1.1.2 Objective function

Often, in reality, each reservoir only has a single turbine. The power output of turbining x , in cubic meters per second m^3/s , is given by a function $\rho(x)$. This function is strictly increasing and concave, i.e., $\rho'(x) \geq 0$ and $\rho''(x) \leq 0$. In our model we have split this range into several subsections (hence several turbines), each with efficiency $\rho_i = \rho'(s_i^*)/3600$ (MWh/m^3) for some s_i^* in each section. We can thus remark that for any two turbines i_1 and i_2 belonging to the same reservoir we have $\rho_{i_1} \geq \rho_{i_2}$ whenever $i_1 \leq i_2$. This approximation comes down to approximating $\rho(x)$ by a piece-wise linear function.

We assume given a time dependent price signal $\lambda(t)$ (in €/MWh). The following objective function has to be minimized, when integrating the cost of used water according to equation (5.1.11):

$$\sum_{n \in \mathcal{N}} \sum_{i=1}^{K_n} (W_i^n (z_{0,i}^n - z_{F,i}^n) - \sum_{t \in \tau} \lambda(t) \Delta t (\sum_{i=1}^{N_T} \rho_i(t) x^i(t) - \sum_{i=1}^{N_P} \frac{1}{\theta_i(t)} y^i(t))), \quad (5.1.21)$$

where, $\theta^i(t)$ is the efficiency of pumping and the auxiliary variables z_i^n satisfy equations (5.1.9),(5.1.10).

5.1.1.3 Matrix formulation

In this section we show that (5.1.4) can be written as bilateral joint chance constraint. This means that the model we are interested in is a bilateral joint chance constrained program with linear objective function and some polyhedral constraints.

Let us consider equation (5.1.3) and apply it recursively to establish the identity

$$\begin{aligned}
V^n(t) &= V^n(0) + \sum_{u=1}^t \sum_{m \in \mathcal{A}(n)} \sum_{i \in \sigma_{\tau}^{-1}[m]} x^i(u - D_m) \Delta t - \sum_{u=1}^t \sum_{i \in \sigma_{\tau}^{-1}[n]} x^i(u) \Delta t \\
&+ \sum_{u=1}^t \sum_{m \in \mathcal{F}(n)} \sum_{i \in \sigma_{\mathcal{P}}^{-1}[m]} y^i(u) \Delta t - \sum_{u=1}^t \sum_{i \in \sigma_{\mathcal{P}}^{-1}[n]} y^i(u) \Delta t \\
&+ \sum_{u=1}^t s_u^n \Delta t + \sum_{u=1}^t \sum_{j=0}^{u-1} \psi_j^n \zeta^n(u - j) \Delta t,
\end{aligned} \tag{5.1.22}$$

holding for all $t \in \tau$ and $n \in \mathcal{N}$. In what follows we will denote with $V_n \in \mathbb{R}^T$ the vector $V_n = (V^n(1), \dots, V^n(T))$. It is of interest to explicitly establish the way in which V_n depends on the vector ζ^n in order to identify the correlation structure of the global underlying uncertainty vector. One easily observes that V_n depends linearly on x and y . In order to establish the correlation structure of the vector ζ , we introduce the matrix mapping $\mathfrak{C} : \mathbb{R}^T \rightarrow \mathcal{M}_{T \times T}$. Here $\mathcal{M}_{T \times T}$ stands for the set of $T \times T$ real matrices and \mathfrak{C} as applied to the sequence $\psi := (\psi_0, \dots, \psi_{T-1}) \in \mathbb{R}^T$ is defined as:

$$\mathfrak{C}(\psi) = \begin{pmatrix} \psi_0 & 0 & 0 & \cdots & 0 \\ \psi_0 + \psi_1 & \psi_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ \sum_{j=0}^{T-1} \psi_j & \cdots & \cdots & \cdots & \psi_0 \end{pmatrix}.$$

It will be convenient to extend the definition of \mathfrak{C} to a sequence ψ shorter than T by appending with zero entries.

Following equation (5.1.22) for each $n \in \mathcal{N}$ we can find a $T \times TN_{\mathcal{T}}$ matrix $M_{\mathcal{T}}^n$ and $T \times TN_{\mathcal{P}}$ matrix $M_{\mathcal{P}}^n$ such that

$$V_n = V_0 - \Delta t M_{\mathcal{T}}^n x + M_{\mathcal{P}}^n y + \Delta t \mathfrak{C}(1) s^n + \Delta t \mathfrak{C}(\psi^n) \zeta^n, \tag{5.1.23}$$

where s^n is the vector formed from the deterministic trend s_t^n of equation (5.1.2). Equations (5.1.9), (5.1.10) can be written easily in linear form by extracting the last line from equation (5.1.23) without the term in ζ^n .

5.1.2 Models for dealing with uncertainty

In this section we will provide our main model, which is a joint chance constrained programming problem (JCCP). We will also provide several alternative models.

5.1.2.1 Expectation model

In a classic version of cascaded reservoir management in short term optimization, uncertainty is assumed to be absent or sufficiently characterized by a forecast. This amounts to

the choice of replacing ζ^n in equation (5.1.23) or equivalently (5.1.2) by its expectation, i.e., $\zeta^n(t) = \mathbb{E}(\zeta^n(t)) = 0 \forall t \in \tau$. This substitution in turn impacts equation (5.1.4).

When combining equations (5.1.9), (5.1.23) and relations (5.1.10), (5.1.5) we know that we can find some extended decision vector (also noted $x \in \mathbb{R}^n$) containing (x, y, z) and some matrix A , vector b such that the system $Ax \leq b$ models all the deterministic constraints (including bounds on x) found in section 5.1.1. One can moreover find a matrix A^r and vectors a^r, b^r such that equation (5.1.4) wherein we have substituted the expectation of ζ for ζ is reflected by $a^r + A^r x \leq 0 \leq A^r x + b^r$.

Combined, this gives the following linear program:

$$\begin{aligned} \min_{x \in \mathbb{R}^n, x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & -A^r x \leq b^r \\ & A^r x \leq -a^r. \end{aligned} \tag{5.1.24}$$

This model can be identified with the model considered in a classical deterministic unit-commitment setting.

5.1.2.2 A Joint Chance Constraint Model (JCCP)

In contrast to the expectation model wherein the effect of uncertainty is neglected, incorporating uncertainty fully in equation (5.1.4) leads to a joint chance constrained program. Indeed, by combining equation (5.1.23) with (5.1.4), we can see that the problem of section 5.1.1 can be cast into the following form, where $\xi \in \mathbb{R}^m$ is a Gaussian random vector with variance-covariance matrix Σ and zero mean (we have explicitly extracted the non-zero average in eq.(5.1.23)):

$$\begin{aligned} \min_{x \in \mathbb{R}^n, x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & p \leq \mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x]. \end{aligned} \tag{5.1.25}$$

In fact the feasible set of (5.1.25) is convex due to the Gaussian character of $\xi \in \mathbb{R}^m$ and a theorem by Prékopa ([181]). This makes the previous optimization problem a convex one. For convenience we define $\varphi : \mathbb{R}^n \rightarrow [0, 1]$ as $\varphi(x) = \mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x]$.

5.1.2.2.1 Link with the Expectation Problem The chance constrained model can be seen as an extension of the expectation model since it takes into account the available stochastic information on the distribution of randomness, whereas model (5.1.24) only uses a single parameter. The following Lemma shows that any feasible solution of (5.1.25) is feasible for (5.1.24). Physically this can be explained by the fact that a "robust" control has to work well in the average situation.

Lemma 5.1.2. *Assume that $p > 0.5$ and that $\xi \in \mathbb{R}^m$ is a symmetric random variable, i.e., $\mathbb{P}[\xi \in A] = \mathbb{P}[\xi \in -A]$ for any measurable set $A \subseteq \mathbb{R}^m$. The feasible set of (5.1.25) is*

contained in the feasible set of (5.1.24). As a consequence the optimal value of (5.1.24) is lower than that of (5.1.25).

Proof. Assume that $x \in \mathbb{R}^n$ is not feasible for (5.1.24), for instance not $a^r + A^r x \leq 0$, i.e., there is at least one strictly positive component. By rearranging we may assume that this is the first one. Now

$\mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x] \leq \mathbb{P}[a^r + A^r x \leq \xi] \leq \mathbb{P}[e_1^\top (a^r + A^r x) \leq e_1^\top \xi] \leq \mathbb{P}[0 < \xi_1] < 0.5$, where ξ_1 is (also) a centered one dimensional symmetric random variable, and e_1 is a standard unit-vector of \mathbb{R}^m . This shows that x can't be feasible for (5.1.25). \square

As mentioned the expectation model is a simple linear program. It is therefore much easier to solve than problem (5.1.25). Despite this fact and the fact that it yields solutions with low optimal values, it will be shown later that the solutions are useless since they violate constraints almost surely.

5.1.2.2.2 An algorithm for solving JCCP In order to solve problem (5.1.25) we will use the supporting hyperplane method. This method was originally introduced by [253] and adapted to the context of joint chance constrained programming by [185, 222]. This algorithm converges in a finite number of steps as shown in [185]. We repeat the algorithm for completeness.

1. (Initialization) Let x_0 be the solution of (5.1.24), x_s a Slater point for (5.1.25). Set $A_0 = A$, $b_0 = b$ and $k = 0$ and pick some tolerance tol , e.g., $tol = 10^{-2}$. Let $\varepsilon > 0$ be a tolerance on the evaluation of φ .
2. (Interpolation) Find λ^* such that $x_k^* = (1 - \lambda^*)x_k + \lambda^*x_s$ and $p - \varepsilon \leq \varphi(x_k^*) \leq p$.
3. (Add Cut) Add constraint $-\nabla\varphi(x_k^*)^\top x \leq -\nabla\varphi(x_k^*)^\top x_k^*$ to the matrix system $A_k x \leq b_k$.
4. (Solve LP) Solve

$$\begin{aligned} \min_{x \in \mathbb{R}^n, x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & A_k x \leq b_k \end{aligned}$$

to find x_{k+1} .

5. (Stopping Test) If $\frac{c^\top(x_k^* - x_{k+1})}{c^\top x_{k+1}} < tol$ then stop, x_k^* is sufficiently optimal, else set $k = k + 1$ and go to step 2.

For the previous algorithm to function we require a Slater point, i.e., some x_s such that $Ax_s \leq b$, and $\varphi(x_s) > p$. It can be obtained by solving the "max-p" problem (see section 5.1.2.3). Moreover, we should be able to efficiently evaluate φ and $\nabla\varphi$. As shown in Lemma 3.2.1 and Corollary 3.2.3, evaluating the gradient can be analytically reduced to computing function values in smaller dimension. Finally computing function values such as $\varphi(x)$ can be done by using the code of Genz ([83]). Evaluating φ and $\nabla\varphi$ requires $2n + 1$ calls to Genz' code.

5.1.2.3 Max-P Problem

We define the "max-p" problem as the following optimization problem:

$$\begin{aligned} \max_{x \in \mathbb{R}^n, x \geq 0} \quad & \varphi(x) := \mathbb{P}[a^r + A^r x \leq \xi \leq b^r + A^r x] \\ & Ax \leq b. \end{aligned} \tag{5.1.26}$$

Clearly any solution x_s of the previous problem with objective function value strictly bigger than p is a Slater point for problem (5.1.25). This "max-p" problem is not only an auxiliary problem for obtaining Slater points, but can also be interpreted as the problem of a decision-maker looking for maximum robustness, regardless of the costs. As a matter of fact if the optimal solution of (5.1.26) is strictly below one, then almost surely satisfying the "random" physical constraints (5.1.4) is not possible. The "max-p" problem therefore also provides us with information on the maximum robustness level p that is "possible".

5.1.2.4 Individual Chance Constraint Model (ICCP)

We consider a simplification of the joint chance constrained model (5.1.25) by transforming each stochastic inequality into individual chance constraints of type $\mathbb{P}[d_1 + \langle a_1, x \rangle \leq \chi] \geq p$ and $\mathbb{P}[\chi \leq \langle a_2, x \rangle + d_2] \geq p$ for well chosen vectors $a_1, a_2 \in \mathbb{R}^n$, scalars d_1, d_2 and a standard Gaussian random variable $\chi \in \mathbb{R}$. An exact formulation is:

$$\begin{aligned} \min_{x \in \mathbb{R}^n, x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & \mathbb{P}[e_i^\top (a^r + A^r x) \leq \xi_i] \geq p \quad \forall i = 1, \dots, m \\ & \mathbb{P}[\xi_i \leq e_i^\top (b^r + A^r x)] \geq p \quad \forall i = 1, \dots, m, \end{aligned} \tag{5.1.27}$$

where $e_i \in \mathbb{R}^m$ is the i -th standard unit vector.

As a matter of fact, model (5.1.27) can be reduced to a simple linear program since the inverse of $F_{\xi_i}(z) = \mathbb{P}[\xi_i \leq z]$ can be evaluated easily. It also offers improved robustness with respect to the expectation model (5.1.24) that offered none. However it can't guarantee a probability level of p for the whole stochastic inequality system and therefore offers far less robustness than the joint model (5.1.25) ([244]). This will become apparent in the numerical experience.

5.1.2.5 A Robust Model

We would like to identify an uncertainty set $\mathcal{E}_p \subseteq \mathbb{R}^m$ for our random inflow process $\xi \in \mathbb{R}^m$ in such a way that the probability of ξ falling in this set is approximately p . We will then enforce the constraints of problem (5.1.25) to hold for all ξ in this set rather than in probability. We will use a specific ellipsoidal form for the uncertainty set and show that the thus obtained robust optimization problem then boils down to a linear

program, once two conic quadratic problems have been solved (see [141] and references therein).

In order to determine \mathcal{E}_p , let $LL^\top = \Sigma$ be the Cholesky decomposition of Σ . Let $y \in \mathbb{R}^m$ be defined as $y = L^{-1}\xi$ and assume that we dispose of a statistical estimate of $\mathbb{E}(y_i^4)$ (in the Gaussian case these are known exactly) for $i = 1, \dots, m$. Whenever the law of ξ is unknown, we can use the variance covariance matrix Σ obtained from statistic estimates. By construction, y is uncorrelated, we will make the (wrong) approximation that this is the same as independence. Now by the Lindeberg-Feller Central Limit Theorem ([189]) we obtain that $y^\top y$ is approximately normally distributed with mean m and standard deviation σ_C , i.e., $y^\top y \approx \mathcal{N}(m, \sigma_C)$, with $\sigma_C = \sqrt{\sum_{i=1}^m \mathbb{E}(y_i^4) - m}$.

We now define $\mathcal{E}_p = \{z \in \mathbb{R}^m : z^\top \Sigma^{-1} z \leq m + \Phi^{-1}(p)\sigma_C\}$. It follows in the case that ξ follows a multivariate Gaussian law that $\mathbb{P}[\xi \in \mathcal{E}_p] = p$. This will be true approximately when ξ follows another multivariate law. This is a very elementary way of deriving an uncertainty set. We refer to [8, 12] for more sophisticated approaches.

We therefore consider the following robust version of problem (5.1.25):

$$\begin{aligned} \min_{x \in \mathbb{R}^n, x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & a^r + A^r x \leq \inf \mathcal{E}_p \\ & b^r + A^r x \geq \sup \mathcal{E}_p, \end{aligned} \tag{5.1.28}$$

where $\inf \mathcal{E}_p \in \mathbb{R}^m$ denotes the vector whose components are the coordinate-wise minima of \mathcal{E}_p ($\sup \mathcal{E}_p \in \mathbb{R}^m$ is defined similarly). Both $\inf \mathcal{E}_p$ and $\sup \mathcal{E}_p$ are solutions of a conic quadratic optimization problem. Indeed model (5.1.28) is equivalent with

$$\begin{aligned} \min_{x \in \mathbb{R}^n, x \geq 0} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & a^r + A^r x \leq \xi \leq b^r + A^r x \quad \forall \xi \in \mathcal{E}_p. \end{aligned}$$

Since model (5.1.28) basically looks at the smallest rectangle containing \mathcal{E}_p and requires satisfaction of constraints for all elements in the rectangle, one could also look at alternative ways to obtain such a rectangle. Basically, we are looking for some $\bar{\xi}$ and $\underline{\xi}$ such that $\mathbb{P}[\underline{\xi} \leq \xi \leq \bar{\xi}] \approx p$. These would then give better bounds than $\inf \mathcal{E}_p$ and $\sup \mathcal{E}_p$ as above, since in general $\mathbb{P}[\inf \mathcal{E}_p \leq \xi \leq \sup \mathcal{E}_p] > p$. In the Gaussian case considered here we can exactly evaluate the probabilistic contents of such rectangles and hence fine-tune the rectangle. Clearly any feasible solution of problem (5.1.25) will also provide such vectors. This last way of obtaining those vectors offers no computational advantage to (5.1.28) other than prematurely ending the algorithm that solves (5.1.25). An alternative would be to take some $q < p$, such that $\mathbb{P}[\inf \mathcal{E}_q \leq \xi \leq \sup \mathcal{E}_q] \approx p$. This is computationally not intensive, but requires evaluations of probabilistic contents. In order to investigate the impact of the choice of this rectangle we have made some runs with model (5.1.28) wherein the rectangle was made to fit perfectly. In practice, we have obtained $\underline{\xi}$ and $\bar{\xi}$ by taking some ad-hoc convex combinations between the Slater point and the solution

of (5.1.24). These results will be referred to as Robust-Calibrated (Robust-Calib) or (5.1.28)-Calib.

5.1.3 Numerical Example

In this section we consider a numerical example from the industry. The instance size is moderate but realistic. The nominal inflows, i.e., s_t^n in equation (5.1.2), are considered constant through time. Finally, the water values are not assumed to depend on the volume, and thereby correspond to the V_0 level. It was shown in section 5.1.1.5 that adding the volume dependency induces no substantial difficulties. The focus of this numerical example is the impact of uncertainty. We will consider 24 time steps of 2 hours each. Figures 4.1(c) and 4.1(a) show further data of our example. This implies the following dimensions for our problem: the Gaussian vector dimension m is 48, the decision vector has 700 elements and the polyhedral constraints are defined by about 1000 linear inequalities.

The system topology, that is the relation between the six power reservoirs in the valley, is expressed by the reservoir connection matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Likewise, the vector $\sigma_{\mathcal{T}} = (1, 1, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 5, 6, 6, 6)$ associates each one of the 16 turbines in the valley to a specific reservoir.

As uncertainty is concerned we will assume that reservoirs 1 ("Vouglaans") and 2 ("Saut Mortier") have random inflows. The standard deviations of the innovations ζ of the inflow process A^n in equation (5.1.2) are taken to be equal to 20% of the nominal inflow values ($0.3 \text{ m}^3/\text{s}$ for reservoir 2). We will consider two instances, one wherein inflows on both reservoirs follow an AR(1) process with coefficient 0.9. A second instance is one wherein we assume that inflows on reservoir 2 follow an AR(3) process with coefficients (0.9, 0.7, -0.7). In this instance inflows on reservoir 1 still follow an AR(1) process. The required probability level p in (5.1.25) is taken to be 0.8.

To express the hydro valley relations in the presence of uncertainty, we need to write down the variance-covariance matrix of the stochastic process. This needs the introduction of some notation. Specifically, we let ψ be a 2×24 matrix such that $\psi_{1,j} = 0.9^j$ and $\psi_{2,j} = 0.9\psi_{2,j-1} + 0.7\psi_{2,j-2} - 0.7\psi_{2,j-3}$, $j = 1, \dots, 24$, where in the last formula any negative indexed terms are assumed 0. We also define the elementary covariance matrix Σ_a :

$$\Sigma_a = \begin{pmatrix} 4.24 & 0 \\ 0 & 0.3 \end{pmatrix} \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \begin{pmatrix} 4.24 & 0 \\ 0 & 0.3 \end{pmatrix}$$

and let Σ_ζ be the block diagonal matrix containing 24 copies of Σ_a . The expression for matrix Σ can be obtained by using matrix $\mathfrak{C}(\psi)$ as defined above. As a result, if R is the 48×48 matrix with $R_{i, \text{mod}(i-1, 24)2+1+\lfloor \frac{i-1}{24} \rfloor} = 1$, it follows that

$$\Sigma = (7200)^2 (R^{-1} \mathfrak{C}(\psi) R) \Sigma_\zeta (R^{-1} \mathfrak{C}(\psi) R)^\top, \quad (5.1.29)$$

is the variance-covariance matrix of the global random inflow vector, that we shall shorten to ξ . The vector ξ is a Gaussian multi-variate random variable in dimension 48.

Solving the problems introduced in Section 5.1.2, we obtain the results as given in Table 5.1, Figures 5.1 and 5.2. We have set a tolerance of 10^{-2} for the supporting hyperplane algorithm for joint chance constrained programming. It should be stated that the true optimal solution of problem (5.1.25) for instance 2 gives a cost, only 0.6% away from the deterministic cost. Indeed the price of chance-constrained robustness is cheap here.

Table 5.1 shows optimal costs and number of violations. In order to compute the latter information, we have made an a posteriori check of empirical probabilities by generating 100 scenarios and counting the number of violations. The volume trajectories resulting from these scenarios are shown in Figure 5.1. Clearly we observe the advantage of using joint chance constrained programming. The additional cost with respect to the deterministic solution is only small, but robustness can be fine tuned. A full robust solution turns out quite costly. Finally individual chance constrained programming can not be used to mimic joint chance constraints as we have no control over the number of violations over a period of time.

Table 5.1: Comparison of costs and number of violations

Inst.	Item	Det (5.1.24)	JCCP (5.1.25)	ICCP (5.1.27)	Robust (\mathcal{E}_p) (5.1.28)-1	Robust (Calib) (5.1.28)-Calib	MaxP (5.1.26)
1	nbViol	100	20	29	0	1	0
1	Cost (€)	$-1.0478e^5$	$-1.0395e^5$	$-1.0443e^5$	$-1.0355e^5$	$-1.0099e^5$	$-9.9176e^4$
2	nbViol	100	20	35	4	21	2
2	Cost (€)	$-1.0478e^5$	$-1.0340e^5$	$-1.0422e^5$	$-1.0282e^5$	$-1.0251e^5$	$-9.9176e^4$

The results of Table 5.1 are readily confirmed by examining the instance of figure 4.1(b) (see also section 4.2.4). We then get:

Table 5.2: Comparison of costs and number of violations

Inst.	Item	Det (5.1.24)	ICCP (5.1.27)	JCCP (5.1.25)	Robust (Calib) (5.1.28)-Calib
Isr48	nbViol	73	34	16	0
Isr48	Cost (€)	$-1.7594e^5$	$-1.7529e^5$	$-1.7504e^5$	$-1.7170e^5$
Isr96	nbViol	91	55	16	2
Isr96	Cost (€)	$-1.7751e^5$	$-1.7649e^5$	$-1.7564e^5$	$-1.5589e^5$

When comparing the turbined volumes in Figure 5.2, one can observe that they are quite similar for most solutions (except for max-p which does not see the cost vector

and is hence only incited to turbine if this allows us to improve robustness) and most reservoirs, except for "Saut Mortier". This reservoir has tight volume bounds and is most heavily impacted by the stochastic inflows. The solution (5.1.25) turbines a bit less in the beginning to avoid violations in time steps 8-10, a bit more during time steps 12-15 to avoid violations there and stops earlier to avoid violations for the last time steps. Solution (5.1.27) offers an intermediate solution. The solution (5.1.28) heavily increases turbining during steps 10-15 and drastically reduces during steps 15-20 for additional robustness. Indeed, even though the uncertainty \mathcal{E}_p is very well calibrated, the solution is over-robust. Unfortunately for larger values of p (in fact $p > 0.85$) this will lead to an empty feasible set of problem (5.1.28), whereas solutions of (5.1.25) can be found. It also shows the difficulty of getting the robust rectangle well calibrated for problem (5.1.28)-Calib. Indeed, even though the rectangle is calibrated to give exactly the same probabilistic contents in both instances, one gives over-robust results (3.6% away from deterministic solution), whereas the other gives more reasonable results as the number of violations is concerned, but still at a large cost (2.2 % away from deterministic solution).

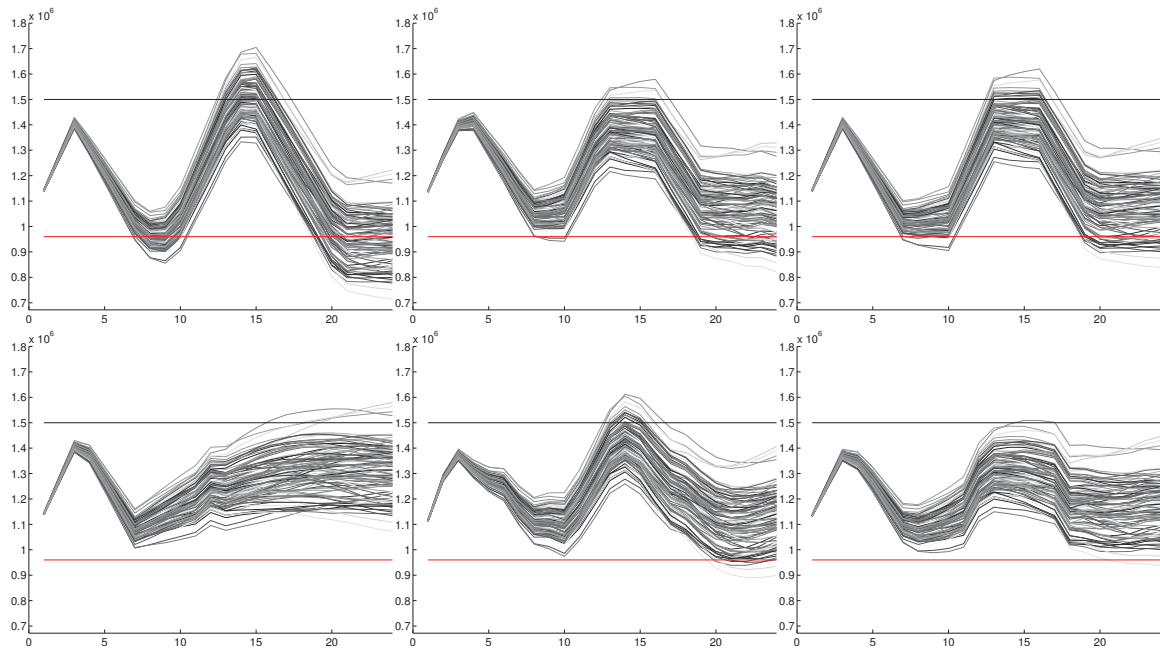


Figure 5.1: Trajectories of filling levels in reservoir "Saut Mortier" and instance 2 for 100 simulated inflow scenarios. From top left to bottom right, solutions of problems (5.1.24), (5.1.25), (5.1.27), (5.1.28), (5.1.28)-Calib and (5.1.26)

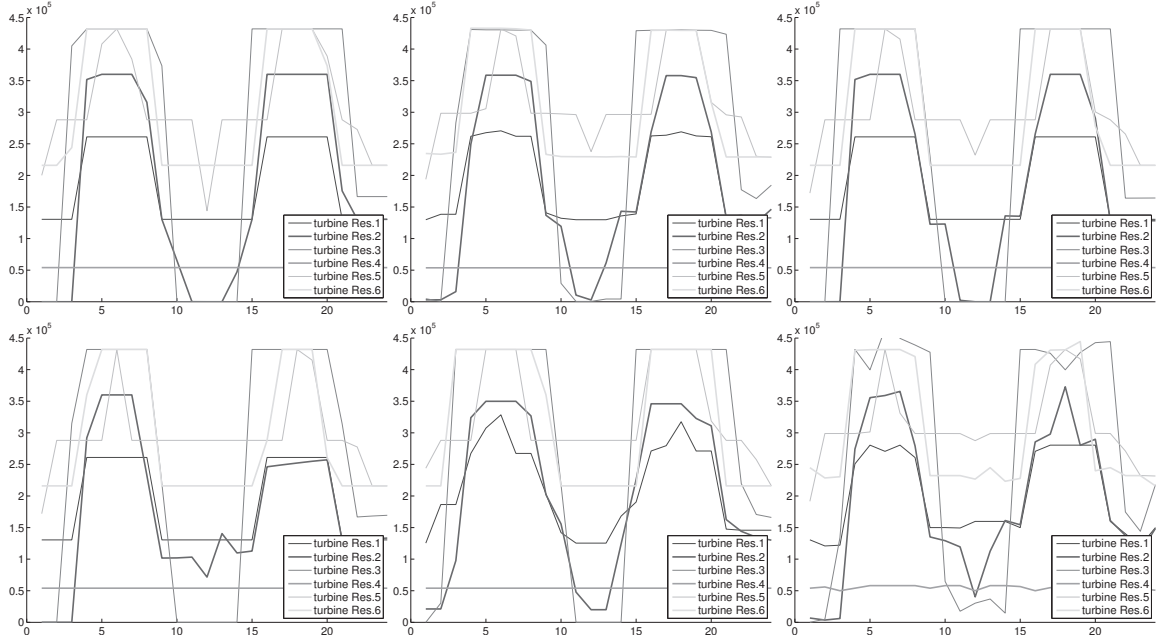


Figure 5.2: Turbined volumes (m^3) for instance 2. From top left to bottom right, solutions of problems (5.1.24), (5.1.25), (5.1.27), (5.1.28), (5.1.28)-Calib. and (5.1.26)

5.2 Robust Unit-Commitment

We begin by recalling that deterministic unit-commitment problems are often decomposed using Lagrangian techniques. Since many unit-commitment problems involve non-convex modelling features, such a Lagrangian decomposition scheme often leads to a non-feasible primary solution. This obtained schedule then needs to be fixed. This can be done by applying heuristics derived from Augmented Lagrangian ([11, 257]) formulations or other techniques (see [210] for some suggestions or [61, 78]). We wish to investigate the effects of using chance constraints for robustifying the production schedule in a unit-commitment framework. As a first step we wish to avoid the complex issues raised by using heuristics as explained above. As a numerical example, we will therefore consider a variant of Unit-Commitment wherein thermal units have a simple (convex) representation. The investigated approaches can clearly be applied unchanged to non-convex Unit-Commitment problems, but the above discussed heuristics might be needed. This will be the object of future work.

Two difficult sub-problems are those related to the thermal generation units ([77]) and the hydraulic valleys (see Section 5.1 above or (for instance) [61] and references therein). Complex dynamic constraints on watershed controls introduce combinatorial aspects in the hydro sub-problem making it even more challenging to solve ([60]).

Ideally, a Robust unit-commitment problem considers at least the following sources of uncertainty:

1. uncertainty on inflows for the hydro reservoirs

2. uncertainty on customer load
3. uncertainty on renewable generation
4. uncertainty on unit availability.

In this section we will consider the first two sources of uncertainty. It can be argued that the third source can be appended to uncertainty of load. It might however be of more interest to consider both sources separately as information on their individual laws might be more readily available than on their aggregated law. Since the deterministic unit-commitment models are already large-scale, their robust counterparts are clearly at least as difficult to solve.

Obtaining a globally robust production schedule means that one has to incorporate uncertainty on load as well. In the literature "stochastic unit-commitment" models are far less common than their deterministic counterparts. Most approaches considered so far use scenario trees in one way or another (e.g. [171, 226, 225, 164, 260]). Uncertainty in each node of the scenario tree is considered known when the decision is taken. When the trunk of the tree covers a whole day (as in [225]) the obtained schedule has seen future uncertainty (past the first day) and can be applied (i.e., sent to the grid operator). Yet uncertainty during the first day is neglected. When the trunk covers less time, it is unclear what schedule has to be sent to the grid-operator ([164]). Recently robust optimization has been applied to stochastic Unit-Commitment ([262, 18]). But the suggested model decouples commitment decisions (starting / stopping status of each unit) from dispatch decisions (power output). The latter decisions are then taken when uncertainty is fully known. Though very interesting, it is unclear still what schedule (including power output) has to be sent to the grid operator. Moreover, in practice, commitment decisions can be adapted (at least to some extent) in a 2nd stage.

Here we make the assumption that both commitment and dispatch decisions have to be taken prior to observing uncertainty. Though it is true that recourse stages exist that allow us to make changes to the previously committed schedule, we do not consider that option in the current model. It is left for future work. We suggest a model that uses a joint-chance constraint to model the deviation between production and load and request that such a deviation remains between bounds with a specific probability. In this way the production schedule becomes robust against uncertainty. Since this chance constraint renders the offer-demand equilibrium constraints highly non-linear (even implicit) it prevents straightforward application of classic decomposition schedules to tackle the large-scale structure. Decomposition schemes are clearly a prerequisite if Robust Unit-commitment is ever to be used in practice. Therefore, we will examine to what extent efficient decomposition schemes can be derived in order to solve the proposed Robust Unit-Commitment problem. We emphasize that our approach differs from the individual chance constrained approaches considered in [56, 167]. Indeed the individual chance constraints on the offer-demand equilibrium with uncertainty on load have an equivalent linear formulation with an additional safety term.

This section is organized as follows: In section 5.2.1 we introduce the specific structure of robust unit-commitment problems. We then propose several decomposition schedules in

section 5.2.2. Some theoretical insights on obtaining Slater points for bilateral joint chance constraints, useful for a numerical solution of joint chance constrained programming, are given in section 5.2.3. Section 5.2.4 contains the description of the model consistent with the highlighted structure. The numeric results obtained when applying the various decomposition schedules and a discussion thereof are provided in section 5.2.6.

5.2.1 Problem Structure

In this section we describe the specific block structure of unit-commitment problems with joint chance constraints. Section 5.2.1.1 contains a short description of our application showing how this structure arises in practice. Section 5.2.1.2 provides a global view of the problem structure.

5.2.1.1 Hydro-Thermal Unit-Commitment Problems

In Hydro-Thermal systems, each production unit generates power to meet some overall equilibrium between customer load and generated power. It is very natural to assume that units generate power without influencing any other units, at least for thermal production units. Clearly hydraulic turbines and pumps influence each other through a network structure and flow constraints. Hydraulic "units" are therefore typically entire hydro valleys for which it becomes reasonable to assume that they can be operated independently. Key constraints are bilateral bounds on volume in each reservoir, flow constraints and potentially complex technical constraints on turbinning/pumping operation. Cost, i.e., the value of water is typically computed by a mid-term planning (optimization) tool, often based on dynamic programming. Since inflows are uncertain and decisions taken before observing uncertainty, one has to come up with a reasonable way to model the above bilateral volume constraints. One way of doing so, is to use chance constrained programming (see Section 5.1). When considering that a Hydro-Valley acts on a price signal, this would imply the following problem structure:

$$\begin{aligned} \min_{x \in \mathbb{R}^k} \quad & c^\top x \\ \text{s.t.} \quad & \mathbb{P}[A^r x + a^r \leq \xi \leq A^r x + b^r] \geq p \\ & Ax \leq b, \end{aligned} \tag{5.2.1}$$

where k is the problem dimension, i.e., the dimension of the decision vector as the hydraulic valley is concerned. Moreover, l_r specifies the number of lines of matrix A^r in the joint probabilistic constraint with safety level p and l_d specifies the number of lines of the matrix A . In the specific case of a hydraulic valley, l_r would typically be the number of reservoirs receiving random inflows times the number of time steps. The size of the parameter l_d would also be linked to various auxiliary variables and related constraints. In applications one has $l_d > l_r$. The vectors a^r, b^r, b, c are of appropriate dimension. The vector $\xi \in \mathbb{R}^{l_r}$ follows some appropriate multi-variate law. In the case of a Hydro-Valley, the cost vector c would reflect costs of using water as computed with loss of opportunity costs and gain as induced by a price signal. The latter price signal might be one coming

from a decomposition schedule. The polyhedral constraints $Ax \leq b$ would contain bounds on turbinng, pumping and bilateral bounds on the volume in reservoirs not impacted by uncertain inflows. The joint chance constraint would deal with bilateral bounds on volume for reservoirs impacted by uncertain inflows. As such the matrix vector product $A^r x$ represents the way in which the volume in each reservoir and for each time step is impacted by the previous turbinng/pumping decisions. Roughly speaking a^r would be the minimal volume bound for each reservoir and time step minus the initial volume and to which we also integrate any deterministically cumulated inflows. The vector b^r is defined similarly but with respect to a maximal volume bound. It is important to note that the joint chance constraint is implicit and highly non-linear. Many assumptions (e.g. ξ Gaussian) can be made under which the feasible set induced by the chance constraint is convex and an appropriate transform of the joint chance constraint a concave function. Then problem (5.2.1) can be either solved by the supporting hyperplane method (see Section 5.2.3) or the bundle methods of Sections 4.1 and 4.2. Again, the main difficulty lies in computing gradients of the joint chance constraint.

As far as the thermal units are concerned, they are subject to complex technical constraints on power variations, starts, ramping rates, minimum up/down times etc... (see [77, 134] for instance). Typical constraints imply that adjacent power levels are sufficiently close (ramping constraints), the unit remains online or offline for a specific amount of time (minimum up/down times), the number of starts is limited, starting costs depend on plant status etc... Most of these constraints imply non-convexities and typical modelling involves binary variables. Therefore, when considering that a thermal unit acts on a price signal, in full generality this would imply a problem structure of the following form:

$$\begin{aligned} \min_{x \in \mathbb{R}^k, z \in \mathbb{Z}^r} \quad & f(x, z) \\ \text{s.t.} \quad & Ax + Bz \leq b, \end{aligned} \tag{5.2.2}$$

where $k + r$ is the dimension of the decision vector for the thermal unit. The objective function f is typically a separable linear or quadratic function and the polyhedral inequality system $Ax + Bz \leq b$ modelling the constraints, contains vectors and matrices of appropriate dimension.

In fact the only constraints that couple the units together are the equilibrium constraints. In a deterministic setting, these simply state that deviation between production and customer load has to remain small. These constraints then have the typical form

$$s^d \leq D - A^l x \leq s^u, \tag{5.2.3}$$

where $s_l, s_u \in \mathbb{R}^T$ are operator chosen bounds, T is the number of time steps in the considered time horizon, D customer load and A^l the $T \times n$ matrix summing up the production of each of the m units aggregated in the decision vector $x = (x_1, \dots, x_m)$. When D is considered to be uncertain and x decided upon before observing D , an appropriate model for (5.2.3) results from requiring the decision vector x to satisfy

$$\mathbb{P}[s^d \leq D - A^l x \leq s^u] \geq p. \tag{5.2.4}$$

A very similar structure is investigated in [202], where stability theory for probabilistically constrained problems is developed. In particular, the authors explicitly consider stability results for probabilistically constrained power dispatch models, showing that the models are stable for several underlying distributions of the load, such as discrete or multi-variate Gaussian. However, no computational results are presented.

Putting these elements together, one ends up with a typical block structure:

$$\begin{aligned} \min_{x:=(x_1,\dots,x_m)\in\mathbb{R}^n} \quad & \sum_{i=1}^m f_i(x_i), \\ \text{s.t.} \quad & x_i \in X_i \subseteq \mathbb{R}^{n_i}, i = 1, \dots, m \\ & s^d \leq D - A^l x \leq s^u, \end{aligned} \tag{5.2.5}$$

where $\sum_{i=1}^m n_i = n$ and the set X_i integrates the constraints of problems (5.2.2) or (5.2.1) depending on whether $i \in \{1, \dots, m\}$ refers to a hydro valley or thermal unit. Finally the mappings $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ associate a cost to each decision vector $x_i \in \mathbb{R}^{n_i}$ for each $i = 1, \dots, m$.

Large-scale (or difficult) instances of problem (5.2.5) arise typically when either the number of units m is very large or the constraint sets X_i are complex for some $i \in \{1, \dots, m\}$. In practice the instances coming from French Unit-commitment problems have both these features and therefore require decomposition methods in order to be solved.

5.2.1.2 A bird's view of the structure

Following the discussion in section 5.2.1.1, unit-commitment problems in Energy management have the following typical abstract structure (see also [61] and references therein):

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & x \in X_1 \cap X_2 \\ & x \in B := \{y : \underline{x} \leq x \leq \bar{x}\}, \end{aligned} \tag{5.2.6}$$

where n is the size of the decision making vector, X_1 is the set modelling all constraints on the individual m units and X_2 the offer-demand equilibrium constraints. This first set is typically structured as a cartesian product of smaller sets. This means that $X_1 = \prod_{i=1}^m X_i^1$, with $X_i^1 \subseteq \mathbb{R}^{n_i}$, $\sum_{i=1}^m n_i = n$. Moreover, the objective function f also allows for a decomposition along the sets X_i^1 , i.e., $f(x) = \sum_{i=1}^m f_i(x_i)$ and $x_i \in X_i^1$ for all $i = 1, \dots, m$. Each of the sets X_i^1 , $i = 1, \dots, m$ roughly contains the feasible production schedules for each unit. In practice a "unit" would be anything ranging from a thermal generation unit, a hydro valley to a specific financial contract.

In many classical applications the set X_2 is a polyhedral set and the sets X_i^1 are defined by many constraints, possibly involving non-linear, non-convex or combinatorial constraints and/or variables.

We will consider uncertainty on both inflows and customer load. This impacts problem (5.2.6) in two ways. Firstly, some of the sets X_i^1 will be defined by additional joint bilateral chance constraints. Secondly, the set X_2 , modelling the offer-demand equilibrium

constraints will no longer be polyhedral (as in (5.2.3)), but defined by a joint bilateral chance constraint itself (as in (5.2.4)). As such, classical decomposition schedules based on Lagrangian decomposition ([138]) can no longer be applied in a straightforward manner. The second difficulty is that even though it has been shown ([246]) that the robust hydro valley sub-problems (those with the joint bilateral chance constraint due to uncertainty on inflows) are solvable, the computational effort required to handle them is still substantial. Any algorithm requiring that computational effort at each iteration/step is incompatible with any operational requirement.

According to the model (5.2.4) it turns out that the problem structure of the problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & x \in X_2, x \in B \end{aligned} \tag{5.2.7}$$

is of the same nature as problem (5.2.1), when f is linear. Indeed the chance constraint (5.2.4) is of the same nature of the chance constraint appearing in problem (5.2.1). Data clearly depends on the specific sub-problem. The convexity of the feasible set of Problem (5.2.1) is of-course important for an efficient solution. In particular whenever the vector $\xi \in \mathbb{R}^{l_r}$ follows a (centered) multi-variate Gaussian law with covariance matrix Σ , convexity can be ensured. Such an assumption is fairly realistic for load as shown in [24]. Indeed, it follows from a classic result by Prékopa (Theorem 4.2.4 [181]) that the feasible set induced by the Probabilistic Constraint is convex in that case. In such a setting, similar algorithms as those found in [246] can therefore be used in order to solve problems (5.2.1).

We will investigate decomposition methods in order to solve (5.2.6) efficiently in section 5.2.2.

5.2.2 Decomposition Methods

We will investigate decomposition methods based on moving the problem structure of problem (5.2.6) artificially into a full cartesian product structure. This can be done by adding additional variables $y \in \mathbb{R}^{\tilde{n}}$ in an appropriate dimension (this technique is called "variable duplication" in [210] and dates back at least to [218]) playing the same role as $x \in \mathbb{R}^n$. An additional constraint is added to enforce their equality. We will speak of "lift", because the problem dimension is increased. Many different lift and decomposition schedules can be set up. They depend on the choice of the dimension \tilde{n} and the way in which we enforce equality with the original problem variables. In this section we will specify several such schedules and investigate their efficiency on a numerical example in section 5.2.6.

5.2.2.1 Lift and Dualize

Applying the previously described "duplication" technique directly, yields the following problem formulation for any $\alpha \in [0, 1]$:

$$\begin{aligned}
 \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^n} \quad & \alpha f(x) + (1 - \alpha)f(y) \\
 \text{s.t.} \quad & x \in X_1 \\
 & y \in X_2 \\
 & x = y \\
 & x \in B, y \in B.
 \end{aligned} \tag{5.2.8}$$

It is easily seen that problem (5.2.8) is equivalent with problem (5.2.6) for all $\alpha \in [0, 1]$. Moreover problem (5.2.8) will admit a Slater point if and only if (5.2.6) admits such points. It does not however exploit any knowledge of the specific nature of the X_2 set. This can be done as explained in section 5.2.2.1.1. The result of which is a lift in smaller dimension, most likely allowing for a more efficient solution.

In order to solve this new problem formulation, we will consider the Lagrangian dual of (5.2.8) where the constraint $x = y$ is relaxed (Duplicating variables + Lagrangian Decomposition is referred to as Lagrangean Decomposition in [93] and references therein). This implies that for each fixed $\mu \in \mathbb{R}^n$, the following primal problem has to be solved:

$$\begin{aligned}
 \min_{x \in \mathbb{R}^n, y \in \mathbb{R}^n} \quad & \alpha f(x) + (1 - \alpha)f(y) + \langle \mu, x - y \rangle \\
 \text{s.t.} \quad & x \in X_1 \\
 & y \in X_2 \\
 & x \in B, y \in B.
 \end{aligned} \tag{5.2.9}$$

This primal problem can be naturally decomposed into sub-problems by unit and an additional load-equilibrium sub-problem. Let $x(\mu), y(\mu)$ denote an arbitrary optimal solution. By construction $x(\mu)$ is feasible for all operational constraints and $y(\mu)$ for the offer-demand equilibrium constraint.

The dual function $\Theta : \mathbb{R}^n \rightarrow \mathbb{R}$ defined as

$$\Theta(\mu) := \alpha f(x(\mu)) + (1 - \alpha)f(y(\mu)) + \langle \mu, x(\mu) - y(\mu) \rangle, \tag{5.2.10}$$

is a concave typically non-differentiable function. We can use a Bundle Method ([114, 20]) in order to maximize (5.2.10), i.e., optimize the problem dual to (5.2.9). It is known that the solution $(x(\mu^*), y(\mu^*))$ at the optimal dual signal μ^* is not necessarily an optimal solution for the primal problem (5.2.9). However, it has been shown that one can take an optimal convex combination of oracle responses (along the iterations) and that this is the optimal solution of the bi-dual problem (see [135, 23]). The convex multipliers are dual variables belonging to the cuts in the last quadratic program solved by the Bundle Method. If problem (5.2.6) is convex, then this approach allows us to obtain the optimal primal solution.

In the specific situation wherein X_1 is a closed convex set, constraint (5.2.4) is dealt with in the form $\log(p) - \log(\mathbb{P}[s^d \leq D - A^l x \leq s^u]) \leq 0$, D follows a non-degenerate multivariate Gaussian distribution, and either f is linear or $\alpha = 1$, it follows from [137, Theorem 4.4] that maximizing (5.2.10) might increase the duality gap with respect to a direct dualization of the probabilistic constraint under the above form. Since, unit-commitment problems generally have a non-convex set X_1 , neither Theorem 4.4 or Theorem 4.5 of [137] can be used and the duality gap may either increase or decrease. Since constraint (5.2.4) is highly non-linear, this might be the price to pay in order to decompose the Lagrangian dual over the m units.

5.2.2.1.1 Dimension Reduction One difficulty in the formulation of problem (5.2.8), and its dual "maximize (5.2.10)" is the dimension of the dual signal μ . Indeed, it is of the global problem dimension n , which might be (very) large. This may in particular imply some difficulties for convergence of non-smooth optimization tools such as Bundle methods in order to maximize (5.2.10). In our setting we know more about the structure of the X_2 set, since $X_2 := \{x \in \mathbb{R}^n : \text{s.t. (5.2.4) holds}\}$. Let T be the size of the random vector D , i.e., $D \in \mathbb{R}^T$. Now it is clear that $x \in X_2$ if and only if $z \in \tilde{X}_2 := \{z \in \mathbb{R}^T : \mathbb{P}[s^d \leq D - z \leq s^u] \geq p\}$ and $A^l x = z$. This alternative description of the set X_2 , called \tilde{X}_2 is a subset of \mathbb{R}^T . Typically T is much smaller than n .

This therefore leads to the following alternative, wherein the problem dimension is equal to the number of lines in the probabilistic offer-demand equilibrium constraint, i.e., T . That formulation is:

$$\begin{aligned} \min_{x \in \mathbb{R}^n, z \in \mathbb{R}^T} \quad & f(x) \\ \text{s.t.} \quad & x \in X_1 \\ & z \in \tilde{X}_2 \\ & A^l x = z \\ & x \in B, z \geq 0, z \leq \bar{z}, \end{aligned} \tag{5.2.11}$$

where \bar{z} is a natural maximum limit on installed capacity of generation and the matrix A^l as defined in equations (5.2.3) and (5.2.4).

The constraint $A^l x = z$ is relaxed and the following dual function considered $\Theta_2 : \mathbb{R}^T \rightarrow \mathbb{R}$:

$$\Theta_2(\mu) := f(x(\mu)) + \langle \mu, A^l x(\mu) - z(\mu) \rangle. \tag{5.2.12}$$

The corresponding dual problem to (5.2.11) is that of maximizing (5.2.12).

5.2.2.2 Lift and Dualize - Augmented Lagrangian

A classic idea to overcome the fact that the primal solution of (5.2.8) at the optimal dual signal μ^* in (5.2.10) is not optimal, is the use of Augmented Lagrangians. A further reason for considering such approaches is the fact that in [9] it is reported that they deal better with a non-convex unit-commitment problem. Non-convexity typically comes from complex constraints on thermal generation.

Considering an Augmented Lagrangian, the primal problem formulation becomes:

$$\begin{aligned}
\min_{x \in \mathbb{R}^n, y \in \mathbb{R}^n} \quad & \alpha f(x) + (1 - \alpha)f(y) + \langle \mu, x - y \rangle + \frac{c}{2} \|x - y\|^2 \\
\text{s.t.} \quad & x \in X_1 \\
& y \in X_2 \\
& x \in B, y \in B,
\end{aligned} \tag{5.2.13}$$

for some appropriate constant $c > 0$. The dual problem would consist of maximizing (5.2.13) over $\mu \in \mathbb{R}^n$.

Unfortunately this problem no longer naturally decomposes directly into sub-problems. One can however apply the auxiliary problem principle ([38, 37]) to obtain an effective decomposition scheme, whenever the problem satisfies appropriate hypotheses. Let $(x_k, y_k) \in X_1 \times X_2$ be a solution generated by an iterative process trying to solve Problem (5.2.13). In order to decompose problem (5.2.13) according to the auxiliary problem principle, let us define the kernel $K^k : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+$ as follows $K^k(x, y) := \frac{\hat{c}}{2} \|x - x_k\|^2 + \frac{\hat{c}}{2} \|y - y_k\|^2$, for some constant $\hat{c} > 0$. Pick a further set of constants $\varepsilon_k > 0$ and let these constants satisfy $\hat{c} > 2\varepsilon_k c$, then this principle shows that solving

$$\begin{aligned}
\min_{x \in \mathbb{R}^n, y \in \mathbb{R}^n} \quad & \varepsilon_k (\alpha f(x) + (1 - \alpha)f(y) + \langle \mu, x - y \rangle) + \frac{\hat{c}}{2} \|x - x_k\|^2 + \\
& + \frac{\hat{c}}{2} \|y - y_k\|^2 + c\varepsilon_k \langle x_k - y_k, x - y \rangle \\
\text{s.t.} \quad & x \in X_1 \\
& y \in X_2 \\
& x \in B, y \in B,
\end{aligned} \tag{5.2.14}$$

to obtain (x_{k+1}, y_{k+1}) will yield a sequence converging to the optimal solution. Moreover the function values will decrease in each iteration. The advantage is that problem (5.2.14) can be decomposed naturally. Moreover μ can be updated by using Uzawa's method, i.e.,

$$\mu_{k+1} = \mu_k + \rho(x_k - y_k), \tag{5.2.15}$$

with an appropriate step size $\rho \in (0, 2c)$. No stopping criteria is exhibited in ([38, 37]) or later documents. We will however choose the stopping criteria $\|x_k - y_k\| < tol$, for some user defined tolerance tol .

5.2.2.2.1 Subproblem Structure When applying the augmented Lagrangian idea, the sub-problem structure is no longer that of (5.2.1), but

$$\begin{aligned}
\min_{x \in \mathbb{R}^k} \quad & c^\top x + \frac{1}{2} x^\top Q x \\
\text{s.t.} \quad & \mathbb{P}[A^r x + a^r \leq \eta \leq A^r x + b^r] \geq p \\
& Ax \leq b,
\end{aligned} \tag{5.2.16}$$

where Q is a $k \times k$ semi-definite matrix (typically even diagonal) and other data is unaltered with respect to problem (5.2.1). In a very similar way as above we also have to add a quadratic term to the objective function of sub-problem (5.2.2).

These sub-problems can be solved by adapting the supporting hyper-planes methods ([181]) idea in a straightforward manner. A similar change can be operated as the Bundle Methods (of Sections 4.1 and 4.2) for solving (5.2.1) are concerned. It however requires solving convex-quadratic programs at each iteration instead of linear programs. Bundle methods already solve such quadratic programs at each iteration, so the overall computational difficulty is not increased.

5.2.2.2 Dimension Reduction In problem (5.2.13), the dual signal μ is again of dimension n , i.e., the global problem dimension. It can be reduced in a very similar way as explained in section 5.2.2.1.1. As such, the Reduced Augmented Lagrangian Problem that we would like to solve becomes:

$$\begin{aligned} \min_{x \in \mathbb{R}^n, z \in \mathbb{R}^T} \quad & f(x) + \langle \mu, A^l x - z \rangle + \frac{c}{2} \|A^l x - z\|^2 \\ \text{s.t.} \quad & x \in X_1 \\ & z \in \tilde{X}_2 \\ & x \in B, z \geq 0, z \leq \bar{z}, \end{aligned} \tag{5.2.17}$$

for an appropriate constant $c > 0$. The dual problem would consist of maximizing (5.2.17) over $\mu \in \mathbb{R}^T$.

Let $(x_k, z_k) \in X_1 \times \tilde{X}_2$ be the current iterate. By picking the Kernel $K^k : \mathbb{R}^n \times \mathbb{R}^T \rightarrow \mathbb{R}_+$ defined as follows $K^k(x, z) := \frac{\hat{c}}{2} \|x - x_k\|^2 + \frac{\hat{c}}{2} \|z - z_k\|^2$, for $\hat{c} > 0$, the auxiliary problem principle requires solving the following problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n, z \in \mathbb{R}^T} \quad & \varepsilon_k f(x) + \langle \mu, A^l x - z \rangle + \frac{\hat{c}}{2} \|x - x_k\|^2 \\ & + \frac{\hat{c}}{2} \|z - z_k\|^2 + c\varepsilon_k \langle (A^l)^\top A^l x_k - (A^l)^\top z_k, x \rangle \\ & + c\varepsilon_k \langle z_k - A^l x_k, z \rangle \\ \text{s.t.} \quad & x \in X_1 \\ & z \in \tilde{X}_2 \\ & x \in B, z \geq 0, z \leq \bar{z}, \end{aligned} \tag{5.2.18}$$

where $\varepsilon_k > 0$ is a set of constants. When the parameters satisfy the inequality $\hat{c} > 2 \max \{\|A\|, 1\} \varepsilon_k c$, solving (5.2.18) yields a sequence (x_{k+1}, z_{k+1}) converging to the optimal solution of (5.2.17). The dual signal $\mu \in \mathbb{R}^T$ can again be updated by applying Uzawa's updating schedule and requires $\rho \in (0, 2c)$ as before. Alternatively one can update it by considering (x_{k+1}, z_{k+1}) as an inexact solution of problem (5.2.17) and using a Bundle Method with inexact oracle (see [44] and reference therein) to update μ .

5.2.2.3 Diagonal Quadratic Approximation

As suggested in [210] we will also investigate the Diagonal Quadratic Approximation (DQA) technique ([208]). It is called a Block Coordinate Descent Method (or non-linear Gauss-Seidel method in [11]). The authors of [11] find it to be far more efficient than the auxiliary problem principle. It's main idea is to solve problems (5.2.13) or (5.2.17) by alternating steps between optimizing over x or y (z), while keeping the other fixed to its current value. In [210] it is suggested to interpret the obtained solution as an inexact solution to (5.2.13) or (5.2.17) and update the dual signal μ according to an inexact Bundle Method. We will therefore follow these suggestions as well and apply the principle on Problem (5.2.17). The impact of using DQA in that setting on the sub-problems (5.2.16) is that the matrix Q is no longer diagonal.

5.2.3 Algorithmic and Numerical Considerations

In this section we provide some Algorithmic and Numerical considerations in order to solve Joint Chance Constrained Programs (JCCP) as (5.2.1) or (5.2.16) as efficiently as possible in order to have efficient decomposition methods. First of all, many algorithms for solving (5.2.1) require the knowledge of a Slater point. Such a point can, in the general case, be computed by applying a very similar solution algorithm to a problem wherein the joint probability is maximized. This is often very involved and computationally quite long, especially when approaching the maximum feasible level. In section 5.2.3.1 we provide some results that allow us to obtain such a point in some cases, by solving a linear program only. Section 5.2.3.2 provides some considerations on how to obtain an efficient decomposition scheme.

5.2.3.1 Slater Points

The key observation in the following Lemmas is that symmetric random variables maximize probability when evaluating symmetric rectangles in bilateral chance constraints.

We recall Definitions 2.5.2 and 2.5.6 and state our key result:

Proposition 5.2.1. *Let $\xi \in \mathbb{R}^m$ be a random variable inducing a symmetric γ -concave probability measure for any $\gamma \in [-\infty, \infty]$. Here symmetric means that $\mathbb{P}[\xi \in A] = \mathbb{P}[(-\xi) \in A]$ for any Lebesgue measurable set A . Then for any $b \in \mathbb{R}^m$ with $b \geq 0$, and $\alpha \in \mathbb{R}^m$ it holds that*

$$\mathbb{P}[-b + \alpha \leq \xi \leq b + \alpha] \leq \mathbb{P}[-b \leq \xi \leq b].$$

Proof. Let $\alpha, b \in \mathbb{R}^m$ with $b \geq 0$ be arbitrary and define $A := \{y \in \mathbb{R}^m : -b \leq y \leq b\}$. It follows from symmetry and the definition of A that

$$\mathbb{P}[-b + \alpha \leq \xi \leq b + \alpha] = \mathbb{P}[\xi \in (A + \alpha)] = \mathbb{P}[\xi \in (A - \alpha)].$$

From Theorem 4.2.4 in [181] it follows that $f(\alpha) := \mathbb{P}[\xi \in (A+\alpha)]$ is a γ -concave function. Now

$$f(0) = f\left(\frac{1}{2}\alpha + \frac{1}{2}(-\alpha)\right) \geq m_\gamma(f(\alpha), f(-\alpha), \frac{1}{2}) = f(\alpha),$$

where the mapping m_γ is as in Definition 2.5.2 proving the Proposition. \square

Corollary 5.2.2. *Let $\xi \in \mathbb{R}^m$ be a random variable having the same properties as in Proposition 5.2.1. Let us consider the following optimization problem*

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & \mathbb{P}[a_r + A_r x \leq \xi \leq b_r + A_r x] \\ \text{s.t.} \quad & Ax \leq b. \end{aligned}$$

Let $\alpha = (b_r + a_r)/2$ and assume that $X := \{x : A_r x + \alpha = 0\} \cap \{x : Ax \leq b\} \neq \emptyset$. Then any $x \in X$ is an optimal solution of the problem.

Proof. Setting $s = (b_r - a_r)/2$ and picking any x one obtains that

$$\mathbb{P}[a_r + A_r x \leq \xi \leq b_r + A_r x] = \mathbb{P}[-s + (A_r x + \alpha) \leq \xi \leq s + (A_r x + \alpha)] \leq \mathbb{P}[-s \leq \xi \leq s],$$

where the latter inequality holds by Proposition 5.2.1 for all x . Moreover we have equality whenever $x \in X$ proving the optimality. \square

Corollary 5.2.3. *Let $\xi \in \mathbb{R}^m$ be a random variable having the same properties as in Proposition 5.2.1. Let us consider the following optimization problem*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & \mathbb{P}[a_r + A_r x \leq \xi \leq b_r + A_r x] \geq p \\ & Ax \leq b. \end{aligned} \tag{5.2.19}$$

Let $\alpha = (b_r + a_r)/2$ and assume that $X := \{x : A_r x + \alpha = 0\} \cap \{x : Ax \leq b\} \neq \emptyset$. Then solving the linear programming problem

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & A_r x = -\alpha, \end{aligned}$$

gives either a Slater point, or an optimal solution for problem (5.2.19) or allows us to conclude on the infeasibility of problem (5.2.19). The latter can be achieved by evaluating the probabilistic constraint on that x .

Proof. By Corollary 5.2.2 it follows that any solution of the linear programming problem maximizes the probability. If this solution yields a probability less than p , problem (5.2.19) is infeasible, bigger than p , it is a Slater point, and equal to p an optimal solution. \square

Remark 5.2.4. *The assumption concerning the set X in Corollary 5.2.2 and 5.2.3 can be easily checked if for instance the matrix A_r is surjective. It then has a right inverse G . Then indeed $x = -G\alpha$ is a solution to the first condition and the set X is not empty if $-AG\alpha \leq b$. If the condition fails to be satisfied, one could look at x minimizing $\|A_r x + \alpha\|_2$ under the constraint $Ax \leq b$. This may turn out to be a reasonable Slater point.*

Remark 5.2.5. *The interest of Corollary 5.2.3 lies in the clear link between problem (5.2.19) and that of our application (5.2.1). One could attempt to use Corollary 5.2.3 or 5.2.2 in order to obtain a Slater point for problem (5.2.1). In our application considered in section 5.2.6 it turns out that Corollary 5.2.3 allows us indeed to obtain a Slater point for both Hydro Reservoir problems and the load sub-problems. This obtained Slater point is not the optimal Solution of problem (5.2.1) and appropriate algorithms still have to be used.*

5.2.3.2 Algorithmic Considerations

In the decomposition schemes proposed in section 5.2.2 each iteration requires the solution of the sub-problems, some of which are joint chance constraint problems. The latter problems are solvable, but can take some time. Now brutally plugging those solution methods inside a decomposition scheme might lead to enormous computation times. But one can do this far more efficiently. Solving such joint chance constrained programming problems can be done by forming a cutting plane approximation of the feasible set (or a cutting plane model for an appropriate transform of the joint probabilistic constraint). If one uses a supporting hyperplane method ([182], see also [246]) in order to solve a joint chance constrained program, one would start with computing the convex combination of the current iterate and the Slater point that exactly meets the required probability level. One would then compute a new cutting plane to improve the current cutting plane model of the constraint set. This cut removes the previous iterate from the solution set. Depending on whether the structure is that of (5.2.1) or (5.2.16) one would solve a linear or quadratic program to come up with the following iterate. If it is feasible one would stop the algorithm. In such a procedure, most time is spent computing the gradients of the joint-chance constraint (i.e., the elements of the Bundle).

When solving problems (5.2.1) or (5.2.16) we thus form a cutting plane model of (a transform of) the joint chance constraint. A change of objective function can be operated and the latter model would remain valid. Since CPU time is spent setting up a cutting plane model of the joint chance constraint, it is clearly of interest to improve this cutting plane model over the course of the iterations of the decomposition schedule. We thus assume that a first cutting plane model is generated and stored to disk (or computed in a first lengthy iteration) for each JCCP sub-problem. It can also be observed that the current solution for a sub-problem solution is feasible for its constraints and can be used as an initial solution for future iterations. Together with the trivial continuity property of Lemma 5.2.6, a small change in the objective function would lead to a near optimal initial solution for two consecutive iterations. Indeed if $x \in \mathbb{R}^k$ is an optimal

solution for a sub-problem inside of a decomposition scheme and the following iteration of this decomposition scheme induces a minor change in the objective function of this sub-problem, then x is nearly optimal for that sub-problem. In particular this means that only few iterations are required in order to appropriately solve the sub-problem. Few iterations imply few costly improvements of the cuttings plane model of the joint chance constraint. Such improvements are of interest only in the vicinity of the global optimal solution.

Lemma 5.2.6. *Let $C \subseteq \mathbb{R}^n$ be a bounded connected set and let $f, g : C \rightarrow \mathbb{R}$ be bounded functions admitting global minima on C . Let $\varepsilon > 0$ be an arbitrary constant and assume that g satisfies $\|g - f\|_\infty < \varepsilon$. Then an optimal solution of the problem $\min_{x \in C} f(x)$ is 2ε optimal for $\min_{x \in C} g(x)$.*

Moreover by solving a linear programming problem using the current cutting planes model, one can obtain a lower bound on the objective function. The current best feasible point, provides an upper bound. We can therefore terminate any solution algorithm for JCCP early if the current best feasible point is sufficiently close (relatively) to the best available bound.

This argument shows that it can be preferable to have small updates in the sub-problem objective functions in order to have fast decomposition iterates.

5.2.4 Numerical Example: a (simplified) Robust Unit-Commitment Problem

5.2.5 Robust System-Dispatch Problem

In this section we will give a detailed description of the simplified Robust Unit-Commitment problem that we will consider in the numerical example. The model considers a Hydro-Thermal Energy mix such as one could encounter in the French system. We will consider a Robust version of a realistic hydro valley model (see Section 5.1 for details) since we consider inflows to be random. We will however consider simplified thermal generation units in order to keep their feasible set convex as the focus is on dealing with robust models as defined by joint chance constraints and not on non-convexities induced by very realistic thermal generation models. Clearly, the suggested methods of section 5.2.2 can easily be applied to models involving far complexer sub-problem models. The applied decomposition method would need to be followed by a Heuristic approach in order to recover an interesting Primal Solution. This is not trivial as shown in [78] and will require substantial additional work. It will be left for future investigations. Finally we will consider randomness on load and suggest a robust model based on joint chance constraints in order to generate sufficient, but not excess energy. The obtained problem is labelled robust as the obtained production schedule protects against uncertainty on inflows and load. The obtained problem structure is exactly that of (5.2.6). The set X_1 will be detailed in sections 5.1 and 5.2.5.1. The set X_2 will be specified in section 5.2.5.2.

We will consider a discretized time horizon. To this end let τ denote the set of (homogeneous) time steps. Let Δt be this time step size expressed in hours. Occasionally T will denote the number of (or last) time steps.

5.2.5.1 The Thermal Generation Unit sub-problem

We will consider a very simple thermal sub-problem, wherein power output of thermal units is considered to be continuous between 0 and some maximum power level P_{mx} . We moreover assume that we have some ramping rate s (MW/h), such that any adjacent power levels may differ no more than $s\Delta t$. Finally thermal units have a cost c_p ($\text{€}/MWh$). With initial power level P_0 and price signal $\lambda \in \mathbb{R}^T$, this gives the following problem.

$$\begin{aligned} \min_{x \in \mathbb{R}^T} \quad & c^\top x \\ \text{s.t.} \quad & \begin{pmatrix} -s\Delta t + P_0 \\ -s\Delta t \\ \vdots \\ -s\Delta t \end{pmatrix} \leq \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix} x \leq \begin{pmatrix} s\Delta t + P_0 \\ s\Delta t \\ \vdots \\ s\Delta t \end{pmatrix}, \end{aligned}$$

where $c = c_p(1, \dots, 1)^\top - \lambda$. This is a linear programming model, and therefore convex.

Some more detailed thermal constraints can be found in [134]. These will however imply a non-convex solution set. We wish to focus only on the difficulties induced by trying to decompose block structured Chance Constrained Programs. As such, for the moment, we will avoid difficulties generated by non-convex thermal sub-problems. Such additional difficulties will be addressed in future work.

5.2.5.2 Offer Demand Equilibrium constraint

We will assume that we have $I^h \cup I^t = \{1, \dots, m\}$ units, where I^h indicates the hydro-valleys and I^t the thermal units. Let \mathbb{R}^{n_i} be the dimension of the sub-problem $i \in \{1, \dots, m\}$, $n_h = \sum_{i \in I^h} n_i$ and $n_t = \sum_{i \in I^t} n_i$. By construction the thermal sub-problems have dimension $n_i = T$. Hydro sub-problems have a decision vector $x_i \in \mathbb{R}^{n_i}$ far larger. By placing the global decision vector x in the order $x = (x_1^h, \dots, x_{|I^h|}^h, x_1^t, \dots, x_{|I^t|}^t) \in \mathbb{R}^{n_h} \times \mathbb{R}^{n_t} = \mathbb{R}^n$ and by doing some bookkeeping one can readily identify the $T \times n$ matrix A^l such that $A^l x$ is the amount of generated power.

It seems reasonable to assume that load $D \in \mathbb{R}^T$ follows a model of the form

$$D_t = \mu_t + \xi_t, \tag{5.2.20}$$

where ξ_t is independent of whatever preceded the first time step t_0 and $t \in \tau$. This implies that $\mathbb{E}(D_t | \mathcal{F}_{t_0}) = \mu_t$ for all $t \in \tau$, where \mathcal{F}_{t_0} is the first element of a filtration $\{\mathcal{F}_t\}_{t \in \tau}$ with which we can equip our probability space. The latter filtration would model information in a dynamic setting, which is not considered here. This makes μ_t the best

load forecast at time t_0 . We moreover assume that ξ_t follows a causal time series model with Gaussian Innovations. This is a fairly realistic assumption as [24] shows. We also assume that we are given some bounds $s^u(t) \geq 0$ and $s^d(t) \leq 0$ that define acceptable load mismatches. Typically one would allow production levels higher than D_t in peak hours and lower than D_t in off-peak hours. The following Joint chance constraint models the offer demand equilibrium constraint:

$$\mathbb{P}[s^d(t) \leq D_t - (A^l x)_t \leq s^u(t) \forall t \in \tau] \geq p, \quad (5.2.21)$$

because of the above assumptions, one can find a Gaussian random variable ξ with variance covariance matrix Σ and 0 mean, such that this constraint is equivalent with

$$\mathbb{P}[s^d + A^l x - \mu \leq \xi \leq s^u + A^l x - \mu] \geq p. \quad (5.2.22)$$

Because of earlier stated results, this implies a convex feasible set for x .

5.2.6 Numerical Example: Data and Results

In this section we provide the results of applying each of the decomposition schedules in section 5.2.2 to the problem detailed in section 5.2.4. Section 5.2.6.1 provides the specific numeric data of this test problem, whereas section 5.2.6.3 provides the numeric results.

5.2.6.1 Numerical Data

We will consider a production park with two hydro valleys, and 9 thermal units. We will consider 24 time steps of 2 hours each. These two hydro valleys are submitted to a joint probability constraints with uncertainty on inflows. Figures 4.1(a) and 4.1(b) provide the hydro valley data. Thermal plant data is given in table 5.3.

This gives a global decision vector dimension of size $n = 1776$. Moreover there are three joint probability constraints, two of which with a random variable in dimension 48 and one with a random variable in dimension 24.

Table 5.3: Thermal Plant data

Data Type	Plants								
Initial Generation	700	700	700	150	150	0	0	0	0
Cost	30	35	37	45	47	60	100	110	150
Max Power	900	900	900	300	300	200	200	200	100
Gradient	100	100	100	30	30	20	20	20	10

5.2.6.2 Numerical results

The numerical results, of which an extensive discussion is provided in section 5.2.6.3 show that Robust Unit-Commitment with joint probabilistic constraints:

1. Can be readily solved to optimality in large scale systems under a convexity assumption. The computational effort is controlled except for a first lengthy iteration in which the cutting plane models for the joint chance constraints have to be set up.
2. Intuitively the augmented Lagrangian approaches with Bundle Methods for the dual form an interesting starting point for the non-convex models.

5.2.6.3 An extensive discussion of Numerical results

In this section we will compare the various decomposition methods suggested in section 5.2.2 on the numeric instance. The goal is to show that an efficient decomposition scheme can be obtained. Two key elements will be examined: feasibility for all constraints of the obtained solution and cost. We will also provide computation time for each method. As observed in section 5.2.3.2 some methods may have lengthy iterations as they can imply "chaotic" changes in the sub-problems objective functions. Especially the robust sub-problem models then require some iterations to converge. Since each such iteration involves the computation of a gradient of a joint-chance constraint, this may take some time.

Since the model of section 5.2.4 is convex by construction, it is known that the lift and dualize method of section 5.2.2.1 converges. Moreover, the optimal convex combination of primal oracle responses in the Lagrangian decomposition (5.2.9) should provide the optimal primal solution. The same holds for the reduced lift and dualize method of section 5.2.2.1.1. We will use a Bundle method to maximize the dual functions (5.2.10) and (5.2.12). The specific Bundle method employed is a variant of the method given in [139] being able to cope with inexact oracles. It turns out that the method of section 5.2.2.1.1 provides an optimal solution indeed. Table 5.4 provides these results:

Table 5.4: Results for the Lift and Dualize Method

Method	objective Value	Nb. Iterations	CPU time (mins)
Reduced L&D 5.2.2.1.1	$3.69963e^6$	280	62.5

It can be noted here that early iterations for maximizing function (5.2.12) imply some iterations for each of the sub-problems, meaning a significant, but not too large change in the objective functions. Later iterations however require almost no further iterations on solving the sub-problems. This means that the Bundle of Information (containing the local joint-chance constraint cuts) is sufficiently rich and only minor changes in the objective function are incurred. This makes sense, since Bundle Methods for maximizing the dual function (5.2.12) will come up with a new signal μ' not far from the current stability center $\hat{\mu}$. It also turns out that it is indeed preferable to keep the dual signal in small dimension, since problem 5.2.2.1 still did not converge after 5000 iterations.

Since the alternative decomposition methods require parameter settings, we can use these results from Lagrangian decomposition to benchmark those decomposition methods. We recall that these alternative methods are considered since they are reported (e.g., in [9])

to work better in a non-convex (deterministic) setting. Table 5.5 provides the results for solving problem (5.2.13) by applying principle (5.2.14).

Table 5.5: Results for the Lift and Dualize - Augmented Lagrangian Methods (The objective function value is shown if a feasible solution is generated)

Method	Obj. Value	Nb. Iter.	CPU (mins)	Parameters $(c, \hat{c}, \rho, \varepsilon)$
AL APP 5.2.2.2	$3.9304e^6$	5000*	484.54	$(1.8e^{-2}, 3.6e^{-2}, 1.8e^{-2}, 1.0)$
AL APP 5.2.2.2	$3.9283e^6$	5000*	593.43	$(0.1, 0.5, 0.1, 1.0)$
AL APP 5.2.2.2	$3.9288e^6$	5000*	475.47	$(1, 5, 0.5, 1.0)$
AL APP 5.2.2.2	$3.9410e^6$	5000*	508.56	$(10, 20, 5, 1.0)$
AL APP 5.2.2.2	-	5000*	553.3	$(1e^{-3}, 2e^{-3}, 1e^{-3}, 1.0)$
AL APP 5.2.2.2	$3.9213e^6$	4646	429.0	$(2e^{-2}, 5e^{-2}, 2e^{-2}, 1.0)$
AL APP 5.2.2.2	$3.9172e^6$	5000*	528.39	$(1, 2, 2e^{-2}, 1.0)$
AL APP 5.2.2.2	$3.9162e^6$	5000*	600.42	$(0.5, 1.0, 0.02, 1.0)$
AL APP 5.2.2.2	$3.9252e^6$	5000*	824.17	$(1.8e^{-2}, 1.8e^{-2}, 1.8e^{-2}, 0.4)$
AL APP 5.2.2.2	$3.9049e^6$	5000*	449.46	$(2e^{-3}, 2e^{-3}, 2e^{-3}, 0.4)$
AL APP 5.2.2.2	$3.9285e^6$	5000*	682.18	$(2e^{-1}, 2e^{-1}, 2e^{-1}, 0.4)$
AL APP 5.2.2.2	-	5000*	455.22	$(2e^{-2}, 2e^{-2}, 3e^{-2}, 0.4)$
AL APP 5.2.2.2	$3.9342e^6$	5000*	689.11	$(2, 2, 2, 0.4)$
AL APP 5.2.2.2	$3.9339e^6$	5000*	527.45	$(2e^{-1}, 2e^{-1}, 2e^{-1}, 0.1)$
AL APP 5.2.2.2	$3.9248e^6$	5000*	524.27	$(2e^{-1}, 2e^{-1}, 1e^{-1}, 0.4)$
AL APP 5.2.2.2	$3.9250e^6$	5000*	541.13	$(2e^{-2}, 2e^{-2}, 1e^{-2}, 0.4)$

One can conclude from table 5.5 that selecting a correct parameter setting for the auxiliary problem principle is delicate. Indeed only one choice of the reported parameters $(c, \hat{c}, \rho, \varepsilon)$ allows the algorithm to terminate before reaching the maximum number of iterations. Nonetheless this principle allows us to ultimately generate a feasible solution for many different parameter settings. These solutions are however approximately 6.2% more costly. Figure 5.3 shows how the augmented Lagrangian term $\|x - y\|_2$ decreases along the number of iterations for a few selected parameter settings of table 5.5.

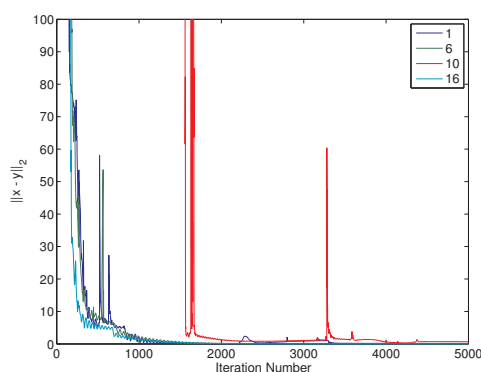


Figure 5.3: Iteration Number vs. $\|x - y\|_2$ for Method 5.2.2.2

In a very similar fashion in which the dimension of the Lagrangian dual problem could be reduced, the same thing can be done for the Augmented Lagrangian technique as shown in section 5.2.2.2.2. Table 5.5 provides numerical results for applying principle (5.2.18) for solving problem (5.2.17).

Table 5.6: Results for the Lift and Dualize - Augmented Lagrangian Methods - Reduced
(The objective function value is shown if a feasible solution is generated)

Method	Obj. Value	Nb. Iter.	CPU (mins)	Parameters ($c, \hat{c}, \rho, \varepsilon$)
AL APP 5.2.2.2.2	-	5000*	300.32	$(2e^{-2}, 4e^{-2}, 2e^{-2}, 1.0)$
AL APP 5.2.2.2.2	-	5000*	280.40	$(2e^{-2}, 2e^{-2}, 2e^{-2}, 0.4)$
AL APP 5.2.2.2.2	-	5000*	407.09	$(2e^{-2}, 5e^{-2}, 2e^{-2}, 1.0)$
AL APP 5.2.2.2.2	-	5000*	349.47	$(2e^{-2}, 1e^{-1}, 2e^{-2}, 1.0)$
AL APP 5.2.2.2.2	-	5000*	323.50	$(2e^{-2}, 5e^{-2}, 2e^{-2}, 0.4)$
AL APP 5.2.2.2.2	-	5000*	318.19	$(0.1, 1.0, 0.05, 1.0)$
AL APP 5.2.2.2.2	-	5000*	376.09	$(4.0, 10.0, 0.5, 1.0)$
AL APP 5.2.2.2.2	-	5000*	331.28	$(40.0, 100.0, 0.5, 1.0)$
AL APP 5.2.2.2.2	-	5000*	355.01	$(4.0, 10.0, 1.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	412.09	$(40.0, 100.0, 10.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	397.44	$(4.0, 10.0, 2.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	345.23	$(40.0, 100.0, 1.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	361.27	$(10.0, 50.0, 2.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	337.46	$(40.0, 100.0, 2.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	406.36	$(4.0, 10.0, 3.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	373.22	$(40.0, 100.0, 3.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	376.28	$(40.0, 100.0, 5.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	stalls	$(400.0, 1000.0, 25.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	450.51	$(4.0, 10.0, 5.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	372.24	$(40.0, 100.0, 6.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	371.4	$(40.0, 100.0, 7.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	405.59	$(4.0, 10.0, 6.0, 1.0)$
AL APP 5.2.2.2.2	-	5000*	368.15	$(60.0, 150.0, 6.0, 1.0)$
AL APP 5.2.2.2.2	$3.9400e^6$	5000*	381.55	$(50.0, 120.0, 6.0, 1.0)$

It turns out that setting up a convergent scheme for the reduced form of the Augmented Lagrangian (5.2.17), decomposed by using the auxiliary problem principle is delicate. Only one set of parameters of the over 50 tested allows us to obtain a feasible solution in the end. Moreover, convergence is very slow compared to the already slow convergence observed for the full version of the Augmented Lagrangian formulation (5.2.13) (decomposed by using the auxiliary problem principle). It seems to be particularly difficult to rightfully set the ρ parameter for Uzawa's updating schedule of the dual signal μ . For this reason we have also attempted to let a Bundle method with inexact oracle update the dual signal. In such a setting we interpret the solution (x_k, z_k) given by the auxiliary problem principle as an inexact solution of (5.2.17). Results are reported in Table 5.7. One can observe that this allows us to generate a feasible solution for all the tested settings $(c, \hat{c}, \varepsilon)$ showing the potential for such an idea.

Table 5.7: Results for the Lift and Dualize - Augmented Lagrangian Methods - Reduced - Updating with Bundle Method (The objective function value is shown if a feasible solution is generated)

Method	Obj. Value	Nb. Iter.	CPU (mins)	Parameters $(c, \hat{c}, \varepsilon)$
AL APP Bdl 5.2.2.2.2	$3.9686e^6$	296	1120.58	(4.0, 10.0, 1.0)
AL APP Bdl 5.2.2.2.2	$4.0390e^6$	83	514.7	(2.0, 5.0, 1.0)
AL APP Bdl 5.2.2.2.2	$3.9321e^6$	194	520.56	(1.0, 3.0, 1.0)
AL APP Bdl 5.2.2.2.2	$3.8576e^6$	287	171.42	(0.5, 2.0, 1.0)

The last suggested decomposition schedule was based on decomposing the augmented Lagrangian formulation (5.2.17) according to the diagonal quadratic approximation and using a Bundle method to update the dual of (5.2.17). Results of applying this suggestion can be found in table 5.8.

Table 5.8: Results for the Diagonal Quadratic Approximation (The objective function value is shown if a feasible solution is generated)

Method	Obj. Value	Nb. Iter.	CPU (mins)	Parameters c
AL DQA 5.2.2.3	$3.89674e^6$	30	321.59	$1e^{-1}$
AL DQA 5.2.2.3	$4.00927e^6$	21	318.26	$2e^{-1}$
AL DQA 5.2.2.3	$3.70072e^6$	208	284.55	$2e^{-2}$

When looking at the results of table 5.8 we again observe the typical additional cost implied by the augmented Lagrangian formulations already observed with the other decomposition schedules. One can also see that when c is small the cost is nearly that of the optimal solution. This result is clearly as expected, since for very small c , the DQA schedule actually boils down to the reduced lagrangian dual decomposition of section 5.2.2.1.1.

In figures 5.4, 5.5 and 5.6 we show the optimal solutions produced by the various decomposition methods.

In figure 5.4 and 5.5 we can see that the optimal turbinning strategies differ quite a lot depending on the decomposition schedule used. It is interesting to observe that

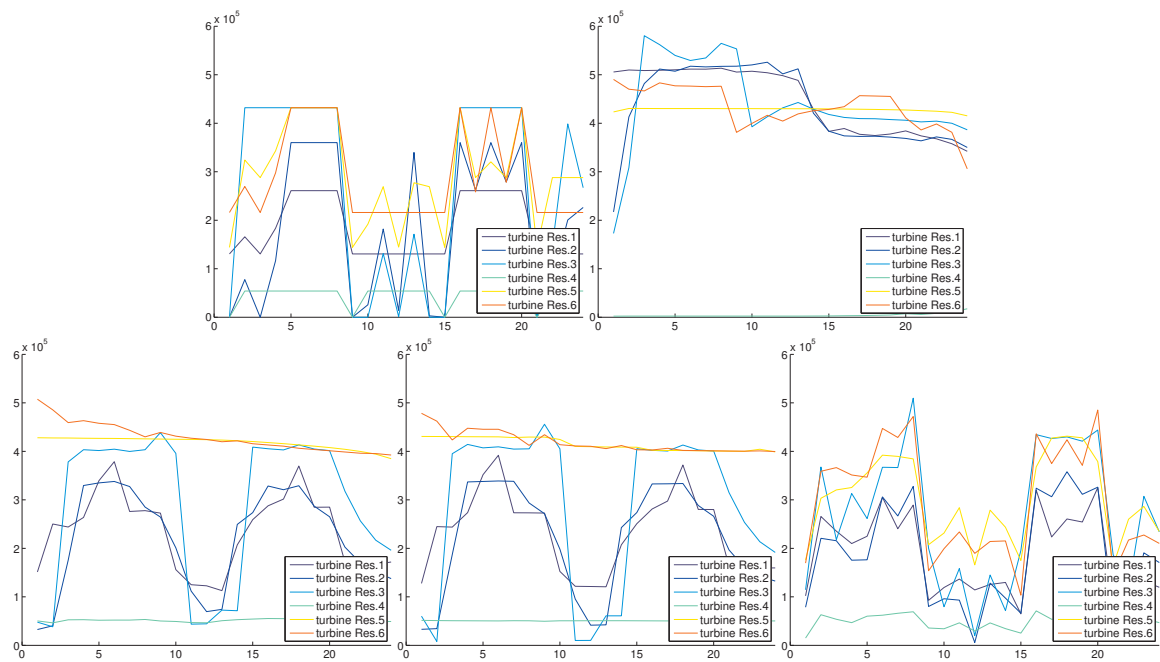


Figure 5.4: Turbined volumes (m^3) for each reservoir of the Ain Valley. From top left to bottom right, solutions produced by Reduced Lagrangian Dual (§5.2.2.1.1), Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2), Reduced Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2.2), Reduced Augmented Lagrangian with Bundle Updating and Reduced Augmented Lagrangian + DQA (§5.2.2.3)

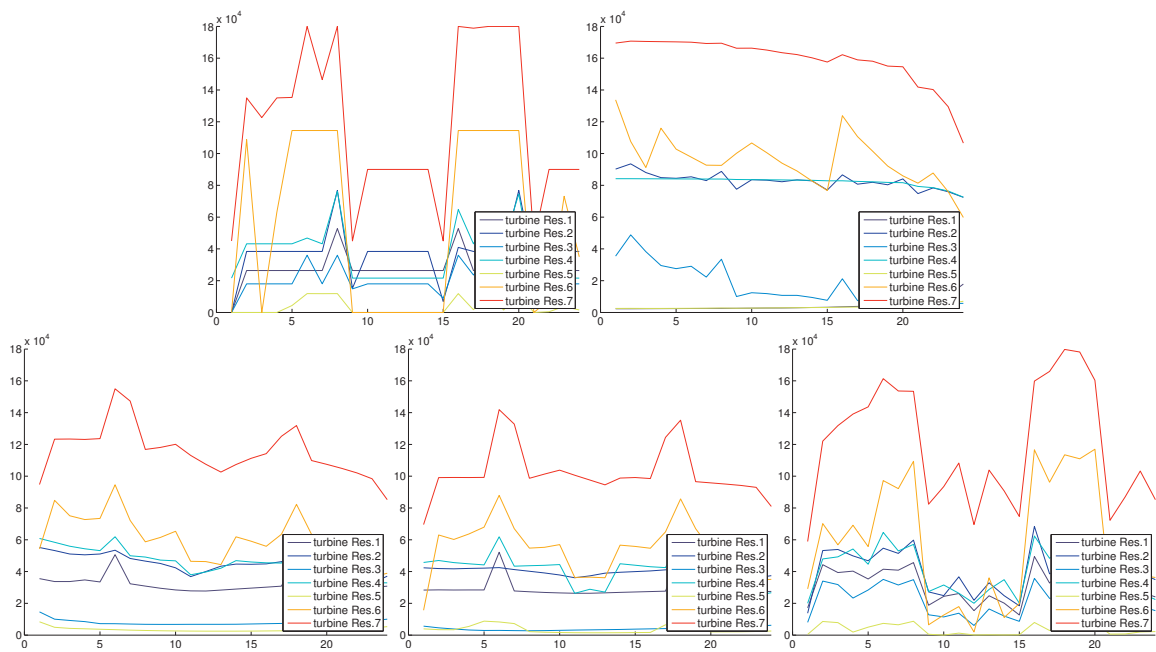


Figure 5.5: Turbined volumes (m^3) for each reservoir of the Isere Valley. From top left to bottom right, solutions produced by Reduced Lagrangian Dual (§5.2.2.1.1), Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2), Reduced Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2.2), Reduced Augmented Lagrangian with Bundle Updating and Reduced Augmented Lagrangian + DQA (§5.2.2.3)

the turbinig strategies produced by the Reduced Lagrangian dual and the Reduced Augmented Lagrangian + DQA are somewhat similar of aspect, whereas those produced by using the auxiliary problem principle are all quite different. One can also see that the solution obtained from using a Bundle method to update the dual signal for the reduced augmented Lagrangian + auxiliary problem principle yields a very similar result when using Uzawa’s updating schedule. The latter having been found quite hard to fine tune, empirically. Figure 5.6 shows that all solutions use the thermal units in quite a similar way.

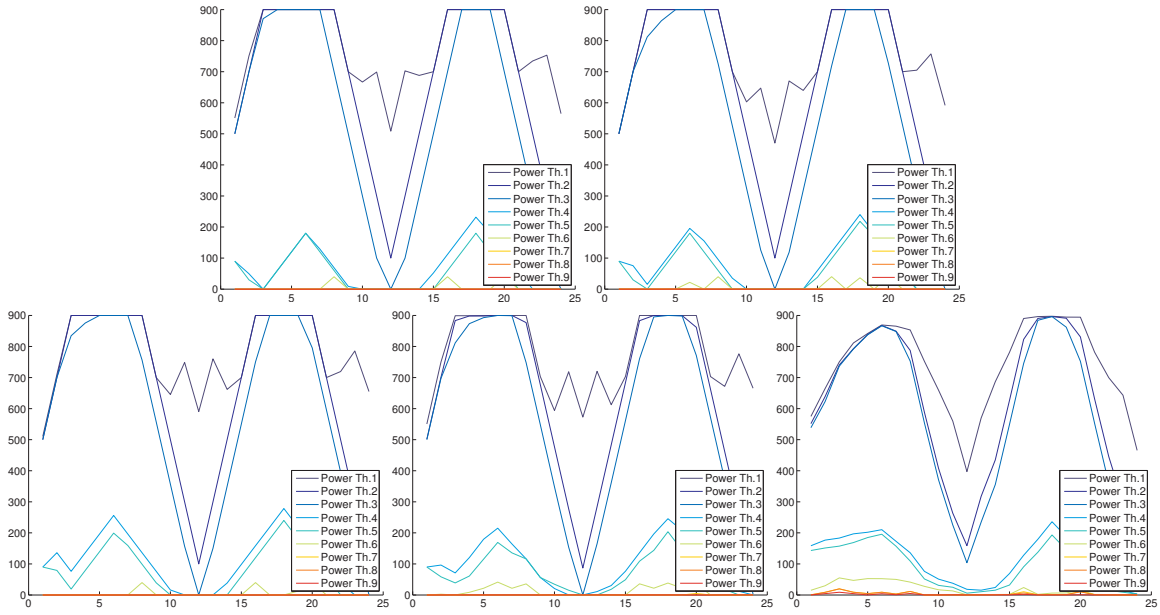


Figure 5.6: Thermal Generation (*MW*) for each thermal unit. From top left to bottom right, solutions produced by Reduced Lagrangian Dual (§5.2.2.1.1), Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2), Reduced Augmented Lagrangian + Auxiliary Problem principle (§5.2.2.2.2), Reduced Augmented Lagrangian with Bundle Updating and Reduced Augmented Lagrangian + DQA (§5.2.2.3)

Finally we can also look at the feasibility for the offer-demand equilibrium constraint (5.2.22). As such we have generated 100 scenarios for the random vector ξ in (5.2.22) and plotted the bounds obtained from the solution for each of the decomposition schedules. These results are shown in figure 5.7. We have also plotted what would happen if we would generate exactly the average load, i.e., $A^l x = \mu$ in (5.2.22). One can observe that the Reduced Lagrangian Dual solution follows closely the general allure of the scenarios and that the lower bound mainly becomes binding near the end of the time horizon. The solutions generated by the augmented Lagrangian formulations and there respective decomposition schedules have some additional bumps that can offer no additional feasibility, since they stick with the RLD solution elsewhere, but do account for the additional cost. Altogether as this aspect is concerned, the different solutions are quite satisfactory.

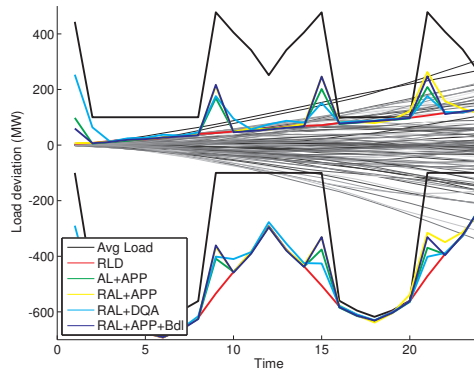


Figure 5.7: Deviations from Load. 100 Scenarios for η in eq.(5.2.22) and lower/upper bounds derived from each of the solutions. Reduced Lagrangian Dual (RLD), Augmented Lagrangian + Auxiliary Problem principle (AL+APP), Reduced Augmented Lagrangian + Auxiliary Problem principle (RAL+APP), Reduced Augmented Lagrangian with Bundle Updating (RAL+APP+Bdl) and Reduced Augmented Lagrangian + DQA (RAL+DQA).

Chapter 6

Concluding Remarks

In this Thesis we have investigated probabilistic programming from a theoretical, algorithmic and applicative point of view. The theoretical work provides new efficient gradient formulae and extends eventual convexity results for specially structured probabilistic constraints. Bundle methods were investigated and applied to probabilistic programming. This led to the definition of the so-called upper oracle, which might provide linearizations that locally overestimate the (convex) function. Rendering unit-commitment robust through the use of probability constraints was the key driver of the applications. In particular we show that realistically sized problems can, in principle, be dealt with.

As perspectives are concerned, the first one involves an extension of the provided gradient formulae to the case wherein we are dealing with a random inequality system having several components. We should examine to what extent those hypothetical results can equally be used for Student (or T -) uncertainty. It is also important to provide counterexamples for differentiability. In particular this implies a thorough investigation of the here formulated growth condition. Finally, it would seem that an alternative derivation would lead to less restrictive conditions for the Student case. Déak's provided sampling scheme could surely also be improved upon and would potentially lead to improved accuracy. This would be of interest for reducing computation time.

As convexity is concerned, we have provided a new insight in the eventual convexity results obtained in [111]. We have shown that those results can be extended to a larger class of Copulae and that sharper probability thresholds can be obtained. However, these results are still largely incomplete. It is of particular interest to examine if some elliptical copulae (Gaussian, T , etc...) exhibit the δ - γ -concavity property as well. Moreover, since Archimedean Copulae decompose as the product of one-dimensional radial distribution and a uniform distribution over the unit-simplex it is readily conjectured that most of these Copulae should be δ - γ -concave. Counter examples should also be looked at to figure out to what extent the provided probability thresholds are tight.

The use of p -efficient points for discrete distributions might allows us to provide conditions under which the feasible set of a joint probabilistic constraint can be approximated by the simpler v -individual probabilistic constraints introduced in this Thesis. When such

positive cases are sufficiently identified it becomes interesting to examine if a separation oracle can be provided. The latter would allow us to identify a violated v -individual constraint for a current candidate solution.

Throughout this work we have remained very attached to the convexity of the feasible set. This is of course important in order to show convergence to a global solution of the optimization problem at hand. However many applications simply do not have convexity at all. Mixed integer programming is a flourishing field in the industry. Unit-commitment problems are modelled in a non-convex way for many years. This shows that convexity might not be as important after all. In order to continue the classic approach of probabilistic programming and considerably extend its applications, convexity could be replaced by local Lipschitz continuity. Such results already exist for multi-variate distribution functions, but would need to be extended to probabilistic constraints involving more general random inequality systems. The gradient can then readily be replaced by a sub-gradient in the sense of Clarke. Bundle algorithms for locally Lipschitz functions already exist and would be a key candidate for dealing with such a non-smooth constraint. It is likely that those methods would need to be extended to constrained optimization with inexact (upper) oracles.

From modelling perspectives, we could integrate the combinatorial constraints on the decision variables, potentially without many difficulties in the cascaded reservoir management problem (e.g. [75, 5]). Also the impact of more realistic models for thermal units (with typically non-convex feasible sets) should be investigated. The investigated decomposition approaches can be applied directly to such a setting, but might need to be followed by heuristics in order to recover an interesting primal solution. Investigating such heuristics offers many research perspectives. We should also consider uncertainty on renewable generation and unit availability. The latter aspect might be particularly tricky as it could induce a non-convex feasible set for the offer-demand equilibrium constraint. Then either we try to deal with such a setting as explained above or we set up a hybrid robust-probabilistically constrained model in order to circumvent the difficulty. Ultimately, a thorough comparison will need to be done in order to investigate the strengths and weaknesses of each approach. This of course includes robust optimization in the sense of Ben-Tal and penalized robust optimization as investigated by [154, 12]. In the latter works, second stage decisions are seen in a simplified manner through the use of a penalty function. The worst case penalty costs are then minimized along with the first stage objective function. It is of particular interest to examine if probabilistically constrained (or robust) optimization can be used instead of a true (robust) multi-stage approach. The advantage of the former over the latter is evidently its improved "tractability".

Bibliography

- [1] L. Andrieu, R. Henrion, and W. Römisch. A model for dynamic chance constraints in hydro power management. *European Journal of Operations Research*, 207:579–589, 2010.
- [2] P. Apkarian, D. Noll, and A. Rondepierre. Mixed H_2/H_∞ control via nonsmooth optimization. *SIAM J. Optim. and Control*, 47(3):1516–1546, 2008.
- [3] R. Apparigliato. *Règles de décision pour la gestion du risque: Application à la gestion hebdomadaire de la production électrique*. PhD thesis, École Polytechnique, Juin 2008.
- [4] V. I. Arkin, A. Shiraev, and R. Wets. *Stochastic Optimization : Proceedings of the International Conference, Kiev 1984*, volume 81 of *Lecture Notes in Control and Information Sciences*. Springer, 1986.
- [5] T. Arnold, R. Henrion, A. Möller, and S. Vigerske. A mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints. *MATHEON-Preprint 1009*, pages 1–14, 2013.
- [6] M. Avriel and I. Zang. *Generalised convex functions with applications to nonlinear Programming. In [251]*. North Holland, 1974.
- [7] F. Babonneau, J.P. Vial, and R. Apparigliato. *Robust Optimization for Environmental and Energy Planning (Chapter 3 in [74])*, volume 138 of *International Series in Operations Research & Management Science*. Springer-Verlag, 2010.
- [8] C. Bandi and D. Bertsimas. Tractable stochastic analysis in high dimensions via robust optimization. *Math. Programming*, 134(1):23–70, 2012.
- [9] J. Batut and A. Renaud. Daily scheduling with transmission constraints: A new class of algorithms. *IEEE Transactions on Power Systems*, 7(3):982–989, 1992.
- [10] A. Belloni, A.L. Diniz, M.E. Maceira, and C. Sagastizábal. Bundle relaxation and primal recovery in unit-commitment problems. the brazilian case. *Annals of Operations Research*, 120(1-4):21–44, 2003.
- [11] C. Beltran and F.J. Heredia. Unit commitment by augmented lagrangian relaxation: Testing two decomposition approaches. *Journal of Optimization Theory and Applications*, 112(2):295–314, 2002.
- [12] S. Ben-Salem. *Gestion Robuste de la production électrique à horizon court-terme*. PhD thesis, Ecole Centrale Paris, Mars 2011.
- [13] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

- [14] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming, series A*, 88:411–424, 2000.
- [15] A. Ben-Tal and A. Nemirovski. On safe tractable approximations of chance-constrained linear matrix inequalities. *Mathematics of Operations Research*, 34(1):1–25, 2009.
- [16] P. Beraldi and A. Ruszczyński. A branch and bound method for stochastic integer problems under probabilistic constraints. *Optimization Methods and Software*, 17:359–382, 2002.
- [17] M. Bertocchi, G. Consigli, and M.A.H. Dempster (Eds). *Stochastic Optimization Methods in Finance and Energy: New Financial Products and Energy Market Strategies*. International Series in Operations Research and Management Science. Springer, 2012.
- [18] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, March 2013.
- [19] P. Billingsley. *Probability and measure*. Wiley Series in Probability and Mathematical Statistics. Wiley InterScience, 3rd edition, 1995.
- [20] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer-Verlag, 2nd edition, 2006.
- [21] C. Borell. Convex set functions in d -space. *Periodica Mathematica Hungarica*, 6:111–136, 1975.
- [22] H.J. Brascamp and E.H. Lieb. On extensions of the Brunn-Minkowski and Prékopa-Leindler theorems, including inequalities for log-concave functions and with an application to the diffusion equations. *Journal of Functional Analysis*, 22:366–389, 1976.
- [23] O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical Programming*, 113(2):299–344, 2008.
- [24] A. Bruhns, G. Deurveilher, and J.S. Roy. A non-linear regression model for mid-term load forecasting and improvements in seasonality. *PSCC 2005 Luik*, 2005.
- [25] J. Bukszár. Hypermultitrees and sharp Bonferroni inequalities. *Mathematical Inequalities and Applications*, 6:727–743, 2003.
- [26] A. Burkauskas. On the convexity problem of probabilistic constrained stochastic programming problems (hungarian). *Alkalmazott Matematikai Lapok*, 12:77–90, 1986.
- [27] G. Calafiore and F. Dabbene (Eds). *Probabilistic and Randomized Methods for Design under Uncertainty*. Springer, 1st edition, 2006.
- [28] G. C. Calafiore and M. C. Campi. Uncertain convex programs: Randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.
- [29] G. C. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Trans. Automat. Control*, 51:742–753, 2006.

- [30] M. C. Campi and S. Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.
- [31] P. Carpentier, G. Cohen, J. C. Culioli, and A. Renaud. Stochastic optimization of unit commitment: a new decomposition framework. *IEEE Transactions on Power Systems*, 11(2), may 1996.
- [32] A. Charnes and W. Cooper. Chance-constrained programming. *Management Science*, 6:73–79, 1959-1960.
- [33] A. Charnes and W. Cooper. Chance constraints and normal deviates. *Journal of the American Statistical Association*, 57:134–148, 1962.
- [34] A. Charnes and W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11:18–39, 1963.
- [35] A. Charnes, W.W. Cooper, and G.H. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4:235–263, 1958.
- [36] E.W. Cheney and A.A. Goldstein. Newton’s method for convex programming and tchebycheff approximation. *Numerische Mathematik*, 1(1):253–268, 1959.
- [37] G. Cohen. Auxiliary problem principle and decomposition of optimization problems. *Journal of optimization Theory and Applications*, 32(3):277–305, 1980.
- [38] G. Cohen and D.L. Zhu. Decomposition-coordination methods in large-scale optimization problems. the non-differentiable case and the use of augmented Lagrangians. *Large Scale Systems, Theory and Applications*, 1, 1983.
- [39] R. Correa and C. Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62(2):261–275, 1993.
- [40] C. d’Ambrosio, A. Lodi, and S. Martello. Piecewise linear approximation of functions of two variables in milp models. *Operations Research Letters*, 38:39–46, 2010.
- [41] Y.S. Davidovich, B.L. Korenblum, and B.I. Hacet. A property of logarithmically concave functions. *Soviet Mathematics Doklady*, 10:477–480, 1969.
- [42] D. P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, August 2004.
- [43] D.P. de Farias and B. Van Roy. Approximate dynamic programming via linear programming. *Advances in Neural Information Processing Systems*, 14, 2002.
- [44] W. de Oliveira and C. Sagastizábal. Level bundle methods for oracles with on demand accuracy. *To appear in Optimization Methods and Software*, pages 1–31, 2014.
- [45] W. de Oliveira, C. Sagastizábal, and C. Lemaréchal. Bundle methods in depth: a unified analysis for inexact oracles. *Available at http://www.optimization-online.org/DB_HTML/2013/02/3792.html*, 2013.
- [46] W. de Oliveira, C.A. Sagastizábal, and S. Scheimberg. Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, 21(2):517–544, 2011.

- [47] W. de Oliveira and M. Solodov. A doubly stabilized bundle method for nonsmooth convex optimization. *Submitted; Preprint available at http://www.optimization-online.org/DB_HTML/2013/04/3828.html*, pages 1–28, 2013.
- [48] I. Deák. *Computer evaluation of a stochastic programming model (in Hungarian)*. PhD thesis, L. Eötvös Univ. of Budapest, 1971.
- [49] I. Deák. Computing probabilities of rectangles in case of multinormal distribution. *Journal of Statistical Computation and Simulation*, 26(1-2):101–114, 1986.
- [50] I. Deák. Subroutines for computing normal probabilities of sets - computer experiences. *Annals of Operations Research*, 100:103–122, 2000.
- [51] D. Dentcheva. *Optimisation Models with Probabilistic Constraints. Chapter 4 in [216]*. MPS-SIAM series on optimization. SIAM and MPS, Philadelphia, 2009.
- [52] D. Dentcheva and G. Martinez. Two-stage stochastic optimization problems with stochastic ordering constraints on the recourse. *European Journal of Operational Research*, 219(1):1–8, 2012.
- [53] D. Dentcheva and G. Martinez. Regularization methods for optimization problems with probabilistic constraints. *Math. Programming (series A)*, 138(1-2):223–251, 2013.
- [54] D. Dentcheva, A. Prékopa, and A. Ruszczyński. Concavity and efficient points for discrete distributions in stochastic programming. *Mathematical Programming*, 89:55–77, 2000.
- [55] P. Diaconis and M. Shahshahani. The subgroup algorithm for generating uniform random variables. *Prob. in Engineering and Info. Sci*, 1:15–32, 1987.
- [56] X. Ding, Wei-Jen Lee, W. Jianxue, and L. Liu. Studies on stochastic unit commitment formulation with flexible generating units. *Electric Power Systems Research*, 80:130141, 2010.
- [57] A.L. Diniz and M.E.P. Maceira. A four-dimensional model of hydro generation for the short-term hydrothermal dispatch problem considering head and spillage effects. *Power Systems, IEEE Transactions on*, 23(3):1298–1308, 2008.
- [58] O. Ditlevsen and H.O. Madsen. *Structural Reliability Methods*. Wiley, 1st edition, 1996.
- [59] J.C. Dodu, M. Goursat, A. Hertz, J.P. Quadrat, and M. Viot. Méthodes de gradient stochastique pour l’optimisation des investissements dans un réseau électrique. *EDF, Bulletin de la Direction des Etudes et Recherches Srie C, Mathmatique, Informatique*, 2:133–167, 1981.
- [60] G. Doukopoulos-Hechmé, S. Charousset-Brignol, J. Malick, and C. Lemaréchal. The short-term electricity production management problem at EDF. *OPTIMA*, 84:2–6, 2010.
- [61] L. Dubost, R. Gonzalez, and C. Lemaréchal. A primal-proximal heuristic applied to french unitcommitment problem. *Mathematical programming*, 104(1):129–151, 2005.
- [62] J. Dupačová. *Stability in stochastic programming - probabilistic constraints. Appears in [4]*, volume 81. Springer, 1986.

- [63] I. Duranyildiz, B. Önöz, and M. Bayazit. A chance-constrained LP model for short term reservoir operation optimization. *Turkish Journal of Engineering*, 23:181–186, 1999.
- [64] M.A.H. Dempster (Ed). *Stochastic Programming*. Academic Press Inc, 1980.
- [65] N.C.P. Edirisinghe, E.I. Patterson, and N. Saadouli. Capacity planning model for a multipurpose water reservoir with target-priority operation. *Annals of Operations Research*, 100:273–303, 2000.
- [66] I. Drietas (Eds). *Stochastic Optimization - Seeing the Optimal for the Uncertain*. INTECH, 2011.
- [67] J. Elstrodt. *Maßund Integrationstheorie*. Springer-Verlag, 7th edition, 2011.
- [68] G. Emiel and C. Sagastizábal. Incremental like bundle methods with applications to energy planning. *Computational Optimization and Applications*, 46(2):305–332, 2009.
- [69] Y.M. Ermoliev, T.Y. Ermolieva, G.J. Macdonald, and V.I. Norkin. Stochastic optimization of insurance portfolios for managing exposure to catastrophic risk. *Annals of Operations Research*, 99:207–225, 2000.
- [70] C.I. Fábián. Bundle-type methods for inexact data. In *Proceedings of the XXIV Hungarian Operations Research Conference (Veszprém, 1999)*, volume 8 (special issue, T. Csendes and T. Rapcsk, eds.), pages 35–55, 2000.
- [71] C.I. Fábián. Computational aspects of risk-averse optimisation in two-stage stochastic models. Technical report, Institute of Informatics, Kecskemét College, Hungary, 2013. Optimization Online report.
- [72] C.I. Fábián and Zoltán Szóke. Solving two-stage stochastic programming problems with level decomposition. *Computational Management Science*, 4:313–353, 2007.
- [73] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Classics in Applied Mathematics. SIAM, 2nd edition, 1987.
- [74] J.A. Filar and A. Haurie. *Uncertainty and Environmental Decision Making: A Handbook of Research and Best Practice*, volume 138 of *International Series in Operations Research & Management Science*. Springer-Verlag, 2010.
- [75] E.C. Finardi and E.L. Da Silva. Solving the hydro unit commitment problem via dual decomposition and sequential quadratic programming. *Power Systems, IEEE Transactions on*, 21(2):835–844, 2006.
- [76] C.A. Floudas and P.M. Pardalos (Eds). *Encyclopedia of Optimization*. Springer - Verlag, 2nd edition, 2009.
- [77] A. Frangioni and C. Gentile. Solving non-linear single-unit commitment problems with ramping constraints. *Operations Research*, 54(4):767–775, 2006.
- [78] A. Frangioni, C. Gentile, and F. Lacalandra. Solving Unit Commitment Problems with General Ramp Constraints. *International Journal of Electrical Power and Energy Systems*, 30:316–326, 2008.
- [79] A. Frangioni, C. Gentile, and F. Lacalandra. Sequential Lagrangian-MILP Approaches for Unit Commitment Problems. *International Journal of Electrical Power*

- and Energy Systems*, 33:585–593, 2011.
- [80] J. Garnier, A. Omrane, and Y. Rouchdy. Asymptotic formulas for the derivatives of probability functions and their Monte Carlo estimations. *European Journal of Operations Research*, 198:848–858, 2009.
- [81] M. Gaudioso, G. Giallombardo, and G. Miglionico. An incremental method for solving convex finite min-max problems. *Math. of Oper. Res.*, 31, 2006.
- [82] A. Geletu, A. Hoffmann, M. Klöppel, and P. Li. Monotony analysis and sparse-grid integration for nonlinear chance constrained process optimization. *Engineering Optimization*, 43:1019–1041, 2011.
- [83] A. Genz. Numerical computation of multivariate normal probabilities. *J. Comp. Graph Stat.*, 1:141–149, 1992.
- [84] A. Genz and F. Bretz. *Computation of multivariate normal and t probabilities*. Number 195 in Lecture Notes in Statistics. Springer, Dordrecht, 2009.
- [85] F. Giannessi, S. Komlósi, and T. Rapcsák (Eds). *New Trends in Mathematical Programming : Hommage to Steven Vajda*, volume 13 of *Applied Optimization*. Springer, 1st edition, 1998.
- [86] P. Girardeau. *Résolution de grands problèmes en optimisation stochastique dynamique et synthèse de lois de commande*. PhD thesis, Université Paris Est, December 2010.
- [87] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Math. Society*, 64:275–278, 1958.
- [88] A. Gouda and T. Szántai. On numerical calculation of probabilities according to dirichlet distribution. *Annals of Operations Research*, 177:185–200, 2010.
- [89] M. Goursat, J.P. Quadrat, and M. Viot. *Stochastic gradient methods for optimizing electrical transportation networks. (Appears in [4])*. Springer, 1986.
- [90] N. Gröwe. Estimated stochastic programs with chance constraints. *European Journal of Operational Research*, 101:285–305, 1997.
- [91] N. Gröwe-Kuska, H. Heitsch, and W. Römisch. Scenario reduction and scenario tree construction for power management problems. *IEEE Bologna Power Tech. Conference*, page 7, 2003.
- [92] J. Guddat, H.Th. Jongen, B. Kummer, and F. Nožička (eds.). *Parametric Optimization and Related Topics (Mathematische Forschung)*. AkademieVerlag, Berlin, 1st edition, 1987.
- [93] M. Guignard. Lagrangean relaxation. *TOP*, 11(2):151–228, 2003.
- [94] S. Gupta. Probability integrals of multivariate normal and multivariate t. *The Annals of Mathematical Statistics*, 34:792–828, 1963.
- [95] H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Computation Optimization and Applications*, 24(2-3):187–206, 2003.
- [96] R. Henrion. *Qualitative stability of convex programs with probabilistic constraints. Chapter 12 in [112]*. Springer Berlin Heidelberg, 2000.
- [97] R. Henrion. On the connectedness of probabilistic constraint sets. *Journal of Optimization Theory and Applications*, 112(3):657–663, 2002.

- [98] R. Henrion. *Perturbation Analysis of Chance-constrained Programs under Variation of all Constraint Data*. Appears in [148]. Springer Berlin Heidelberg, 2004.
- [99] R. Henrion. Structural properties of linear probabilistic constraints. *Optimization: A Journal of Mathematical Programming and Operations Research*, 56(4):425–440, August 2007.
- [100] R. Henrion. Optimierungsprobleme mit wahrscheinlichkeitsrestriktionen: Modelle, struktur, numerik. *Lecture Notes*, page 43, 2010.
- [101] R. Henrion. Gradient estimates for Gaussian distribution functions: Application to probabilistically constrained optimization problems. *Numerical Algebra, Control and Optimization*, 2:655–668, 2012.
- [102] R. Henrion. *A Critical Note on Empirical (Sample Average, Monte Carlo) Approximation of Solutions to Chance Constrained Programs (Chapter 3 in [116])*. Springer, 2013.
- [103] R. Henrion, C. Küchler, and W. Römisch. Discrepancy distances and scenario reduction in two-stage stochastic integer programming. *Journal of Industrial and Management Optimization*, 4:363–384, 2008.
- [104] R. Henrion, C. Küchler, and W. Römisch. Scenario reduction in stochastic programming with respect to discrepancy distances. *Computational Optimization and Applications*, 43:67–93, 2009.
- [105] R. Henrion and A. Möller. Optimization of a continuous distillation process under random inflow rate. *Computer & Mathematics with Applications*, 45:247–262, 2003.
- [106] R. Henrion and A. Möller. A gradient formula for linear chance constraints under Gaussian distribution. *Mathematics of Operations Research*, 37:475–488, 2012.
- [107] R. Henrion and W. Römisch. Metric regularity and quantitative stability in stochastic programs with probabilistic constraints. *Mathematical Programming*, 84:55–88, 1999.
- [108] R. Henrion and W. Römisch. Hölder and lipschitz stability of solution sets in programs with probabilistic constraints. *Mathematical Programming*, 100:589–611, 2004.
- [109] R. Henrion and W. Römisch. Lipschitz and differentiability properties of quasi-concave and singular normal distribution functions. *Annals of Operations Research*, 177:115–125, 2010.
- [110] R. Henrion and C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41:263–276, 2008.
- [111] R. Henrion and C. Strugarek. *Convexity of Chance Constraints with Dependent Random Variables: the use of Copulae*. (Chapter 17 in [17]). Springer New York, 2011.
- [112] N. Van Hien, J-J. Strodiot, and P. Tossings (Eds). *Optimization : Proceedings of the 9th Belgian-French-German Conference on Optimization Namur*, volume 481 of *Lecture Notes in Economics and Mathematical Systems*. Springer Berlin Heidelberg, 2000.
- [113] M. Hintermüller. A proximal bundle method based on approximate sub-

- gradients. *Computational Optimization and Applications*, 20:245–266, 2001. 10.1023/A:1011259017643.
- [114] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Number 306 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 2nd edition, 1996.
- [115] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- [116] D. Hömberg and F. Tröltzsch (Eds). *System modeling and Optimization*, volume 391 of *IFIP Advances in Information and Communication*. Springer-Verlag, 2013.
- [117] M. Jüger and D. Naddef (Eds). *Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions*. Lecture Notes in Computer Science. Springer-Verlag, 2001.
- [118] P. Kall. *On approximations and stability in stochastic programming. Appears in [92]*. Akademie-Verlag, 1st edition, 1987.
- [119] P. Kall and J. Mayer. *Stochastic Linear Programming: Models, Theory and Computation*. International Series in Operations Research and Management Science. Springer, 1st edition, 2005.
- [120] E. Karas, A. Ribeiro, C. Sagastizábal, and M. Solodov. A bundle-filter method for nonsmooth convex constrained optimization. *Math. Program.*, 116(1):297–320, June 2008.
- [121] S. Kataoka. A stochastic programming model. *Econometrica*, 31:181–196, 1963.
- [122] V. Kaňková. A note on estimates in stochastic programming. *Journal of Computational and Applied Mathematics*, 56:97–112, 1994.
- [123] J.E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.
- [124] A.I. Kibzun and Y.S. Kan. *Stochastic Programming Problems with Probability and Quantile Functions*. Wiley, 1st edition, 1996.
- [125] A.I. Kibzun and V.Y. Kurbakovskiy. Guaranteeing approach to solving quantile optimization problems. *Annals of Operations Research*, 30:81–94, 1991.
- [126] A.I. Kibzun and S. Uryas’ev. Differentiability of probability function. *Stoch. Anal. Appl.*, 16:1101–1128, 1998.
- [127] K.C. Kiwiel. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Math. Programming*, 69(1):89–109, 1995.
- [128] K.C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on Optimization*, 16(4):1007–1023, 2006.
- [129] K.C. Kiwiel. A method of centers with approximate subgradient linearizations for nonsmooth convex optimization. *SIAM Journal on Optimization*, 18(4):1467–1489, 2008.
- [130] K.C. Kiwiel. An inexact bundle approach to cutting stock problems. *Inform. Journal on Computing*, 22:131–143, 2010.
- [131] K.C. Kiwiel. Bundle methods for convex minimization with partially inexact ora-

- cles. *To appear in Comp. Opt. Appl*, 2012.
- [132] K.C. Kiwiel and C. Lemaréchal. An inexact bundle variant suited to column generation. *Math. Program.*, 118(1):177–206, 2009.
 - [133] R. Lang. A note on the measurability of convex sets. *Archiv des Mathematik*, 47(1):90–92, 1986.
 - [134] N. Langrene, W. van Ackooij, and F. Bréant. Dynamic constraints for aggregated units: Formulation and application. *IEEE transactions on Power Systems*, 26(3):1349–1356, 2011.
 - [135] C. Lemaréchal. *Lagrangian relaxation*. In [117] (Chapter 4). Springer-Verlag, 2001.
 - [136] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Math. Programming*, 69(1):111–147, 1995.
 - [137] C. Lemaréchal and A. Renaud. A geometric study of duality gaps, with applications. *Math. Programming*, 90:399–427, 2001.
 - [138] C. Lemaréchal and C. Sagastizábal. An approach to variable metric bundle methods. *Lecture Notes in Control and Information Science*, 197:144–162, 1994.
 - [139] C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: from conceptual to implementable forms. *Mathematical Programming*, 76(3):393–410, 1997.
 - [140] J. Lindenstrauss and V.D. Milman (Eds). *Geometric aspects of functional analysis : Israel Seminar (GAFA) 1989-1990*, volume 1469 of *Lecture Notes in Mathematics*. Springer-Verlag Berlin, 1991.
 - [141] M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.
 - [142] H.A. Loiaciga. On the use of chance constraints in reservoir design and operation modeling. *Water Resources Research*, 24:1969–1975, 1988.
 - [143] D. P. Loucks, J. R. Stedinger, and D. A. Haith. *Water Resource Systems Planning and Analysis*. Prentice-Halls, Inc., 1981.
 - [144] J. Luedtke. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming*, To Appear:1–26, 2013.
 - [145] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19:674–699, 2008.
 - [146] K. Marti. Differentiation of probability functions : The transformation method. *Computers and Mathematics with Applications*, 30:361–382, 1995.
 - [147] K. Marti. Differentiation of probability functions : The transformation method. *Math. Programming*, 75(2):201–220, 1996.
 - [148] K. Marti, Y. Ermoliev, and G. Pflug (Eds). *Dynamic Stochastic Optimization*, volume 532 of *Lecture Notes in Economics and Mathematical Systems*. Springer Berlin Heidelberg, 2004.
 - [149] J. Mayer. *On the Numerical solution of jointly chance constrained problems. Chapter 12 in [238]*. Springer, 1st edition, 2000.
 - [150] A. McNeil and J. Nešlehová. Multivariate archimedian copulas, d-monotone functions and l_1 norm symmetric distributions. *The Annals of Statistics*, 37:3059–3097,

- 2009.
- [151] M. Meyer and S. Reisner. *Characterization of affinely-rotation invariant log-concave measures by section-centroid location. (in [140])*. Lecture Notes in Mathematics. Springer-Verlag Berlin, 1991.
 - [152] B.L. Miller and H.M. Wagner. Chance constrained programming with joint constraints. *Operations Research*, 13:930–945, 1965.
 - [153] M. Minoux. *Programmation Mathématique: Théorie et Algorithmes*. Tec & Doc Lavoisier, 2nd edition, 2007.
 - [154] M. Minoux. Solving some multistage robust decision problems with huge implicitly defined scenario trees. *Algorithmic Operations Research*, 4(1):1–18, 2009.
 - [155] D.R. Morgan, J.W. Eheart, and A.J. Valocchi. Aquifer remediation design under uncertainty using a new chance constraint programming technique. *Water Resources Research*, 29:551–561, 1993.
 - [156] S. Nadarajah and S. Kotz. Mathematical properties of the multivariate t distribution. *Acta Applicandae Mathematicae*, 89:53–84, 2005.
 - [157] S.G. Nash and A. Sofer. A barrier-method for large-scale constrained optimization. *ORSA Journal on Computing*, 5:40–53, 1993.
 - [158] S.G. Nash and A. Sofer. Why extrapolation helps barrier methods. Technical report, Department of Operations Research and Engineering, George Mason University, 1998.
 - [159] A. Nemirovski and A. Shapiro. Scenario approximations of chance constraints. *Preprint* : http://www.optimization-online.org/DB_HTML/2004/11/1000.html, pages 1–45, 2004.
 - [160] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal of Optimization*, 17(4):969–996, 2006.
 - [161] A. Nemirovski and A. Shapiro. *Scenario Approximations of Chance Constraints (Chapter 1 in [27])*. Springer, 2006.
 - [162] O. Nilsson and D. Sjelvgren. Variable splitting applied to modelling of start-up costs in short term hydro generation scheduling. *Power Systems, IEEE Transactions on*, 12(2):770–775, 1997.
 - [163] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 2nd edition, 2006.
 - [164] M.P. Nowak and W. Römisch. Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. *Annals Of Operations Research*, 100(1-4):251–272, December 2000.
 - [165] M.P. Nowak, R. Schultz, and M. Westphalen. A stochastic integer programming model for incorporating day-ahead trading of electricity into hydro-thermal unit commitment. *Optimization and Engineering*, 6:163–176, 2005. 10.1007/s11081-005-6794-0.
 - [166] N.J. Olieman and B. van Putten. Estimation method of multivariate exponential probabilities based on a simplex coordinates transform. *J. Stat. Comput. Simul.*, 80:355–361, 2010.

- [167] U.A. Ozturk, M. Mazumdar, and B.A. Norman. A solution to the stochastic unit commitment problem using chance constrained programming. *IEEE Transactions on Power Systems*, 19(3):1589–1598, August 2004.
- [168] B. Pagnoncelli, S. Ahmed, and A. Shapiro. Sample average approximation method for chance constrained programming: Theory and applications. *J. Optim. Theory Appl.*, 142:399–416, 2009.
- [169] M.V.F. Pereira and L.M.V.G. Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical Programming*, 52(2):359–375, 1991.
- [170] G. Pflug and H. Weisshaupt. Probability gradient estimation by set-valued calculus and applications in network design. *SIAM J. Optim.*, 15:898–914, 2005.
- [171] A.B. Philpott, M. Craddock, and H. Waterer. Hydro-electric unit commitment subject to uncertain demand. *European Journal of Operational Research*, 125:410–424, 2000.
- [172] A.B. Philpott, A. Dallagi, and E. Gallet. On cutting plane algorithms and dynamic programming for hydroelectricity generation. Available online at <http://www.epoc.org.nz/papers/MORGANEvsDOASA.pdf>, pages 1–20, 2011.
- [173] A.B. Philpott and Z. Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.
- [174] R.A. Ponrajah, J. Witherspoon, and F.D. Galiana. Systems to optimise conversion efficiencies at ontario hydro’s hydroelectric plants. *Power Systems, IEEE Transactions on*, 13(3):1044–1050, 1998.
- [175] A. Prékopa. On probabilistic constrained programming. *Proceedings of the Princeton Symposium on Math. Prog.*, 28:113–138, 1970.
- [176] A. Prékopa. Logarithmic concave measures with applications to stochastic programming. *Acta Scientiarum Mathematicarum (Szeged)*, 32:301–316, 1971.
- [177] A. Prékopa. A class of stochastic programming decision problems. *Matematische Operations forschung und Statistik*, 3:349–354, 1972.
- [178] A. Prékopa. On logarithmic concave measures and functions. *Acta Scientiarum Mathematicarum (Szeged)*, 34:335–343, 1973.
- [179] A. Prékopa. Programming under probabilistic constraints with a random technology matrix. *Mathematische Operationsforchung und Statistik*, 5:109–116, 1974.
- [180] A. Prékopa. Sharp bound on probabilities using linear programming. *Operations Research*, 38:227–239, 1990.
- [181] A. Prékopa. *Stochastic Programming*. Kluwer, Dordrecht, 1995.
- [182] A. Prékopa. *Probabilistic programming*. In [209] (Chapter 5). Elsevier, Amsterdam, 2003.
- [183] A. Prékopa, S. Ganczer, I. Deák, and K. Patyi. *The STABIL stochastic programming model and its experimental application to the electrical energy sector of the Hungarian economy*. In [64]. Academic Press Inc., 1980.
- [184] A. Prékopa, T. Rapcsák, and I. Zsuffa. Serially linked reservoir system design using stochastic programming. *Water Resources Research*, 14:672678, 1978.
- [185] A. Prékopa and T. Szántai. Flood control reservoir system design using stochastic

- programming. *Math. Programming Study*, 9:138–151, 1978.
- [186] A. Prékopa and T. Szántai. A new multivariate gamma distribution and its fitting to empirical streamflow data. *Water Resources Research*, 14:19–24, 1978.
- [187] A. Prékopa and T. Szántai. On optimal regulation of a storage level with application to the water level regulation of a lake. *European Journal of Operations Research*, 3:175–189, 1979.
- [188] A. Prékopa, B. Vízvári, and T. Badics. *Programming under probabilistic constraints with discrete random variable*. In [85]. Springer, 1998.
- [189] Yu.V. Prokhorov and V. Statulevičius (Eds.). *Limit Theorems of Probability Theory*. Springer-Verlag, 2000.
- [190] E. Raik. Qualitative research into the stochastic nonlinear programming problems (russian). *Eesti NSV Teaduste Akademia Toimetised*, 20:8–14, 1971.
- [191] E. Raik. The quantile function in stochastic nonlinear programming (russian). *Eesti NSV Teaduste Akademia Toimetised*, 20:229–231, 1971.
- [192] E. Raik. On the stochastic programming problem with the probability and quantile functionals (russian). *Eesti NSV Teaduste Akademia Toimetised*, 21:142–148, 1972.
- [193] E. Raik. The differentiability in the parameter of the probability function and optimization of the probability function via the stochastic pseudogradient method (russian). *Izvestiya Akad. Nauk Est. SSR, Phis. Math.*, 24(1):3–6, 1975.
- [194] T. Rapcsák. *On the Numerical Solution of a Reservoir Model (in Hungarian)*. PhD thesis, University of Debrecen (Hungary), 1974.
- [195] E.Ya. Remez. Sur un procédé convergent d’approximations successives pour déterminer les polynomes d’approximation. *C.R. Acad. Sci. Paris*, 198:2063–2065, 1934.
- [196] Y. Rinott. On the convexity of measures. *Annals of Probability*, 4:1020–1026, 1976.
- [197] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1st edition, 1970.
- [198] R.T. Rockafellar and S. Uryas’ev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.
- [199] R.T. Rockafellar and S. Uryas’ev. Conditional value at risk: optimization approach. *Stochastic optimization: algorithms and applications*, pages 411–435, 2001.
- [200] N. Roenko. *Stochastic programming problems with integral functionals over multi-values mappings (in Russian)*. PhD thesis, USSR Kiev, 1983.
- [201] W. Römisch. *Stability of Stochastic Programming Problems*. In [209] (Chapter 8). Elsevier, Amsterdam, 2003.
- [202] W. Römisch and R. Schultz. Distribution sensitivity for certain classes of chance-constrained models with application to power dispatch. *Journal of Optimization Theory and Applications*, 71:569–588, 1991.
- [203] W. Römisch and R. Schultz. Stability analysis for stochastic programs. *Annals of Operations Research*, 30:241–266, 1991.
- [204] W. Römisch and R. Schultz. Stability of solutions for stochastic programs with complete recourse. *Mathematics of Operations Research*, 18:590–609, 1993.
- [205] C. Roos, T. Terlaky, and J-P. Vial. *Interior Point Methods for Linear Optimization*.

- Springer-Verlag New York, 2nd edition, 2005.
- [206] J.O. Royset and E. Polak. Implementable algorithm for stochastic optimization using sample average approximations. *Journal of Optimization Theory and Applications*, 122(1):157–184, 2004.
 - [207] J.O. Royset and E. Polak. Extensions of stochastic optimization results to problems with system failure probability functions. *Journal of Optimization Theory and Applications*, 133(1):1–18, 2007.
 - [208] A. Ruszczyński. On convergence of an augmented lagrangian decomposition method for sparse convex optimization. *Mathematics of Operations Research*, 20(3):634–656, 1995.
 - [209] A. Ruszczyński and A. Shapiro. *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, 2003.
 - [210] C. Sagastizábal. Divide to conquer: Decomposition methods for energy optimization. *Mathematical Programming*, 134(1):187–222, 2012.
 - [211] C. Sagastizábal and M. Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM Journal on Optimization*, 16(1):146–169, 2005.
 - [212] G. Salinetti. Approximations for chance-constrained programming problems. *Stochastic*, 10:157–179, 1983.
 - [213] I.J. Schoenberg. On Pólya frequency functions I. the totally positive functions and their laplace transforms. *Journal Analyse Mathématique*, 1:331–374, 1951.
 - [214] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2:121–152, 1992.
 - [215] A. Shapiro. *Monte Carlo sampling methods*. Chapter 6 in [209], volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier, 2003.
 - [216] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming. Modeling and Theory*, volume 9 of *MPS-SIAM series on optimization*. SIAM and MPS, Philadelphia, 2009.
 - [217] R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications*. Springer, 1st edition, 2000.
 - [218] R. Soenen. *Contribution à l'étude des systèmes de conduite en temps réel en vue de la commande d'unités de fabrication*. PhD thesis, University of Lille, 1977.
 - [219] M.V. Solodov. On approximations with finite precision in bundle methods for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 119(1):151–165, October 2003.
 - [220] Y. Song, J. Luedtke, and S. Küçükyavuz. Chance constrained binary packing problems. *Preprint* : http://www.optimization-online.org/DB_FILE/2012/10/3639.pdf, pages 1–23, 2013.
 - [221] T. Szántai. *Numerical evaluation of probabilities concerning multi-dimensional probability distributions*. PhD thesis, Hungarian Academy of Sciences, 1985.
 - [222] T. Szántai. A computer code for solution of probabilistic-constrained stochastic

- programming problems. In (*Y. Ermoliev and R.J.-B. Wets eds.*): *Numerical Techniques for Stochastic Optimization*, pages 229–235, 1988.
- [223] T. Szántai. Evaluation of a special multivariate gamma distribution. *Mathematical Programming Study*, 27:1–16, 1996.
- [224] T. Szántai. Improved bounds and simulation procedures on the value of the multivariate normal probability distribution function. *Annals of Operations Research*, 100:85–101, 2000.
- [225] S. Takriti, J.R. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 11:1497–1508, 1996.
- [226] S. Takriti, B. Krasenbrink, and L.S.Y. Wu. Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. *Operations Research*, 48(2):268–280, April 2000.
- [227] E. Tamm. On g -concave functions and probability measures (russian). *Eesti NSV Teaduste Akademia Toimetised, Füüsika-Matemaatika*, 28:17–24, 1977.
- [228] D.M. Topkis and A.F. Veinott. On the convergence of some feasible direction algorithms for nonlinear programming. *SIAM Journal on Control*, 5(2):268–279, 1967.
- [229] P. Torrion and J. Leveugle. Comparaison de différentes méthodes d’optimisation appliquées à la gestion annuelle du système offre-demande d’électricité français. *M.85.108*, page 20, Juillet 1985.
- [230] A. Turgeon. Optimal operation of multi-reservoir power systems with stochastic inflows. *Water Resources Research*, 16(2):275–283, 1980.
- [231] S. Uryas’ev. Differentiation formula for integrals over sets given by inclusion. Technical Report WP-88-59, IIASA, 1988.
- [232] S. Uryas’ev. On the differentiability of an integral over the set given by inclusion (in russian). *Kibernetika*, 5:83–86, 1988.
- [233] S. Uryas’ev. A differentiation formula for integrals over sets given by inclusion. *Numerical functional Analysis and Optimization*, 10(7&8):827–841, 1989.
- [234] S. Uryas’ev. Derivatives of probability functions and integrals over sets given by inequalities. *Journal of Computational and Applied Mathematics*, 56(1-2):197–223, 1994.
- [235] S. Uryas’ev. Derivatives of probability functions and some applications. *Annals of Operations Research*, 56:287–311, 1995.
- [236] S. Uryas’ev. *Introduction to the Theory of Probabilistic Functions and Percentiles (Value-at-Risk) Chapter 1 in [238]*. Kluwer Academic Publishers, 2000.
- [237] S. Uryas’ev. *Derivatives of probability and Integral functions: General Theory and Examples. Appearing in [76]*. Springer - Verlag, 2nd edition, 2009.
- [238] S. Uryas’ev (ed). *Probabilistic Constrained Optimization: Methodology and Applications*. Kluwer Academic Publishers, 2000.
- [239] W. van Ackooij. Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit-commitment. *Submitted ; Preprint CR-2012-08*, 2012.

- [240] W. van Ackooij. Eventual convexity of chance constrained feasible sets. *To Appear in Optimization (A Journal of Math. Programming and Operations Research)*, pages 1–22, 2013.
- [241] W. van Ackooij and W. de Oliveira. Level bundle methods for constrained convex optimization with various oracles. *to Appear in Computation Optimization and Applications; Preprint available at http://www.optimization-online.org/DB_FILE/2013/05/3891.pdf*, pages 1–33, 2013.
- [242] W. van Ackooij and R. Henrion. Gradient formulae for nonlinear probabilistic constraints with gaussian and gaussian-like distributions. *Draft submitted, WIAS Preprint No. 1799*, pages 1–18, 2013.
- [243] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Early evaluation of chance-constrained programming for energy management optimization problems. Technical Report H-R36-2010-00447-EN, EDF R&D, 2010.
- [244] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. On probabilistic constraints induced by rectangular sets and multivariate normal distributions. *Mathematical Methods of Operations Research*, 71(3):535–549, 2010.
- [245] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. *Chance Constrained Programming and Its Applications to Energy Management*. In [66] (Chapter 13). IN-TECH, 2011.
- [246] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *to Appear in Optimization and Engineering*, 2011.
- [247] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. On joint probabilistic constraints with Gaussian Coefficient Matrix. *Operations Research Letters*, 39:99–102, 2011.
- [248] W. van Ackooij and C. Sagastizábal. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *to Appear in Siam Journal on Optimization; Preprint available at http://www.optimization-online.org/DB_HTML/2012/12/3711.html*, pages 1–25, 2012.
- [249] W. van Ackooij and R. Zorgati. Estimating the probabilistic contents of gaussian rectangles faster in joint chance constrained programming for hydro reservoir management. *EngOpt2012, 3rd International Conference on Engineering optimization*, July 2012.
- [250] C. van de Panne and W. Popp. Minimum-cost cattle feed under probabilistic protein constraints. *Management Science*, 9:405–430, 1963.
- [251] P. van Moeseke (Eds). *Mathematical Programming for activity Analysis*. North Holland, Amsterdam, 1974.
- [252] R.J. Vanderbei and D.F. Shanno. Interior-point algorithms for nonconvex nonlinear programming: orderings and higher-order methods. *Math. Programming*, 87:303–316, 2000.
- [253] A.F. Veinott. The supporting hyperplane method for unimodal programming. *Operations Research*, 15:147–152, 1967.

- [254] B. Vízvári. The integer programming background of a stochastic integer programming algorithm of Dentcheva-Prékopa-Ruszczynski. *Optimization Methods and Software*, 17:543–559, 2002.
- [255] S.W. Wallace and S.-E. Fleten. Stochastic programming models in energy (chapter 10 in [209]). In A. Ruszczynski and A. Shapiro, editors, *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 637–677. Elsevier, 2003.
- [256] J. Wang. Continuity of feasible solution sets of probabilistic constrained programs. *Journal of Optimization Theory and Applications*, 63:79–89, 1989.
- [257] S.J. Wang, S.M. Shahidehpour, D.S. Kirschen, S. Mokhtari, and G.D. Irisarri. Short-term generation scheduling with transmission and environmental constraints using an augmented lagrangian relaxation. *IEEE Transactions on Power Systems*, 10(3):1294–1301, August 1995.
- [258] S.S. Wilks. *Mathematical Statistics*. John Wiley & Sons, 1962.
- [259] S.J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, 1st edition, 1987.
- [260] L. Wu, M. Shahidehpour, and T. Li. Stochastic security-constrained unit commitment. *IEEE Transactions on Power Systems*, 22(2), May 2007.
- [261] V.A. Zalgaller. Mixed volumes and the probability of falling into convex sets in case of multivariate normal distributions. *Math. Zametki (in Russian)*, 2:97–104, 1967.
- [262] L. Zhao and B. Zeng. Robust unit commitment problem with demand response and wind energy. In *Proceedings of IEEE Power and Energy Society General Meeting, 2012*, 2012.
- [263] R. Zorgati and W. van Ackooij. Optimizing financial and physical assets with chance-constrained programming in the electrical industry. *EngOpt2008, International Conference on Engineering optimization*, 2008.
- [264] R. Zorgati and W. van Ackooij. Optimizing financial and physical assets with chance-constrained programming in the electrical industry. *Optimization and Engineering*, 12(1):237–255, 2011.
- [265] R. Zorgati, W. van Ackooij, and R. Apparigliato. Supply shortage hedging: estimating the electrical power margin for optimizing financial and physical assets with chance-constrained programming. *IEEE Transactions on Power Systems*, 24(2):533–540, 2009.
- [266] G. Zoutendijk. *Methods of Feasible Directions : a Study in Linear and Non-Linear Programming*. Elsevier, 1st edition, 1960.